

webMethods Advanced Troubleshooting Guide

Version 4.1

December 2012

Software AG Global Support

Copyright © 2005-2012 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.

Statement of Conditions

SOFTWARE AG. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SOFTWARE AG BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OR DATA, INTERRUPTION OF BUSINESS, OR FOR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, EVEN IF SOFTWARE AG HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THIS PUBLICATION OR IN THE WEBMETHODS SOFTWARE.

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

Introduction	5
Integration Server - General Troubleshooting	9
Integration Server - Performance	15
Integration Server - Hang	20
Integration Server - Crash	26
Integration Server - Startup	32
Integration Server - Shutdown	36
Integration Server - ART	37
Integration Server - Cluster	39
Integration Server - Pub/Sub	42
Integration Server - Reverse HTTP Gateway	48
Integration Server - Scheduler	51
Integration Server - Session	53
Broker - General Troubleshooting.....	56
Broker - Documents.....	79
Broker - Crash.....	80
Broker - Territory	85
Adapters - JDBC Adapter.....	88
Adapters - SAP Adapter	97
Adapters - Siebel Adapter	102
Adapters - Siebel Connection Sample Code	106
Adapters - Websphere MQ Adapter	109
B2B products - Trading Networks	114
B2B products - C1 Onramp Adapter.....	121
B2B products - EDI	123
B2B products - EDIINT Module	125
B2B products - RosettaNet Module.....	129
B2B products - SWIFT Module	132
BAM Products - Optimize.....	138

BPM Products - CAF	143
BPM Products - Designer	146
BPM Products - Monitor	149
BPM Products - My webMethods Server	157
BPM Products - Process Engine	162
BPM Products - Task Engine	176
SOA Products - CentraSite	180
SOA Products - Tamino	183
SOA Products - Insight	190
Other products - Deployer	199
Other products - Developer	202
Other products - Software AG Installer	205
Other products - MDM OneData	209
Platform-wide - Disclaimer	218
Platform-wide - How to obtain a JVM Thread Dump	219
Platform-wide - CPU	221
Platform-wide - Memory	226
Platform-wide - Network	231
Platform-wide - SSL	234
Platform-wide - Database	248
Platform-wide - Dr. Watson Logs	251
Index	253

Introduction

Overview

Welcome to the latest version of the Advanced Troubleshooting Guide for the webMethods platform. This guide is created by the Global Support team at Software AG to share some of our top troubleshooting tips, procedures, and lessons learned during troubleshooting customer issues.

The objective of the Advanced Troubleshooting Guide is to provide guidance on troubleshooting development and run time problems a webMethods user might encounter with the webMethods Integration Platform. This guide also provides recommendations regarding information to provide to Global Support when submitting a Support Incident in order to expedite troubleshooting and resolution of the problem.

This guide is not an 'official' component of webMethods product documentation. It is not subject to the same quality controls and review processes as is the official webMethods Software documentation. The philosophy of this guide is to put in writing the knowledge contained in the minds of webMethods support engineers, and to document the steps we often guide customers through during the initial stages of a Support Incident. Feedback is welcome and can be sent to GS-WEBM-ATG@softwareag.com.

Audience

This guide is intended for developers, testers, project managers, and any other users involved in the implementation and administration of the webMethods Integration Platform.

Scope

The webMethods Advanced Troubleshooting Guide provides information about major webMethods components, including the Integration Server, My webMethods Server, Trading Networks, Monitor, Designer et cetera. We plan to cover additional areas in future releases. Unless otherwise noted, the information in this guide applies to all the versions of webMethods products starting with the 7.1 platform. Some information may still be applicable to older versions, however this is not guaranteed.

General Guidelines For All Products

Service Pack/Fix Level

You can check for availability of service packs and fixes for webMethods products on **Empower**, the official Software AG customer extranet, at

<http://empower.softwareag.com>

- To find Service Packs: Products -> Product Version Availability
- To find Fixes: Knowledge Center -> Product Fixes

System administrators and others responsible for ongoing maintenance of a webMethods implementation should use Empower to subscribe to service pack and release announcements, and regularly check for fixes that are applicable to their environments.

Please read the usage notes sections in all readme files and the known limitations section, if available, to understand the usage and impact of the fix.

Also, please read the "Special Consideration" section in the installation guide carefully to get acquainted with the any special considerations associated with installing the product on different platforms.

Checking Which Service Packs and Fixes Are Applied

Integration Server

- Open the Integration Server Administrator (typically located at <http://hostname:5555>).
- Click "About" in the upper right-hand corner. Fixes and service packs are listed on the "Updates" line under the "Software" heading.
 - A more detailed page is shown with the "View" link on the "Packages/Updates" line under the "Server Environment" heading.

My webMethods Server

- Open the My webMethods Server Home Page (typically located at <http://localhost:8585>).
- Click "About" in the upper right-hand corner. Fixes and service packs are listed on the "My webMethods Server" tab under the "Licensed Products" heading.

Broker

- Run the following command from a command line prompt to display the version number of the Broker Server:

```
<webMethods>/Broker/bin/awbroker -version
```

General Guidelines

Most production incidents with webMethods software are found, during post-mortem, to have been avoidable through proper planning. Factors that are key to a trouble-free production environment are:

Good design

Understanding the behavior and inter-dependencies of the components that make up the Integration Platform is essential to good design. Be aware of known limitations and design your solutions accordingly. For specific principles and best practices relating to the design of a webMethods solution, consult the webMethods Community. Communities are located on <http://communities.softwareag.com/>.

Testing

Proper testing is the most important factor towards achieving a trouble-free production system. Key points to keep in mind:

- Test systems should be similar to those used in the production environment and should be available at all times. To the closest extent possible, formal staging environments should exactly duplicate production environments in terms of operating systems, CPUs, memory, hard disk, running applications, and so forth. Systems used for testing during development should also resemble the production environment as much as possible. If your production system is running servers in a cluster, the test system should also be clustered in the same way.
- Test every individual component of a solution thoroughly before migrating the solution to production.
- Be sure to test under various load conditions. At a minimum, test your solution under both typical and peak volumes. If possible, you should also perform testing under two to three times the typical and peak volumes.
- Before applying a fix or service pack to a production system, perform thorough regression tests in a staging environment. Back up key system areas before applying fixes and service packs.
- Back up all mission-critical data regularly. An effective backup strategy is very important for disaster recovery. Test your disaster recovery plan.
- If you have a reproducible case, please document the test case step-by-step.

Contact Global Support

The service we provide is most effective if you provide us with as much information as possible about your question or problem when you contact us. When calling in, please have the following information at hand. This will help us in resolving your support issue swiftly.

Support Incident details

- Your Software AG Support Incident number, if you are contacting us regarding a previously reported issue.
- Your Software AG customer number.
- Version, service pack/fix level of your Software AG products.
- Versions of all related software, including operating system, network components, etc.
- Fixes/patches which have been applied to your Software AG products and any changes that have been made in your environment recently.
- Specific error numbers and/or messages, where applicable.
- Sequence of events that led to the error.

Business Impact

Please classify the severity of your case according to the following definitions:

- **CRISIS** - Severe production outage
- **CRITICAL** - Significant production business impact
- **STANDARD** - Visible error or issue

When you open a new product support incident, we will ask you questions concerning the problem you are experiencing. Based on your description of the problem and your assessment of the impact on your business, the Support Consultant will provide assistance in determining the severity level applied to your incident.

Integration Server - General Troubleshooting

Diagnostic Utility

You can use the handy diagnostic utility to collect configurational, operational, and logging data from the Integration Server in one shot. You can invoke the following service from Integration Server using Administrator credentials:

```
http://(host):(port)/invoke/wm.server.admin/getDiagnosticData
```

The diagnostic utility creates a temporary `diagnostic_data.zip` file in the `IntegrationServer\logs` directory. When you invoke this service from the browser, you will be prompted with a dialog box to save `diagnostic_data.zip` to the desired location.

The following files are included in `diagnostic_data.zip` (tested in IS 7.1.2):

- `config\JDBCpools.txt`
- `config\Ports.txt`
- `config\Server.cnf`
- `config\ServerAbout.txt`
- `config\ServicesInfo.txt`
- `config\TN_Properties.cnf`
- `config\TradingNetworks.txt`
- `logs\auditxxxxxxx.log`
- `logs\auditactivityxxxxxxx.log`
- `logs\auditassocxxxxxxx.log`
- `logs\errorxxxxxxx.log`
- `logs\processxxxxxxx.log`
- `logs\sessionxxxxxxx.log`
- `logs\stats.log`
- `logs\txinxxxxxxx.log`
- `logs\txoutxxxxxxx.log`
- `runtime\JDBCpools.txt`
- `runtime\Ports.txt`
- `runtime\Scheduler.txt`
- `runtime\ServerStats.txt`
- `runtime\ServiceUsage.txt`
- `runtime\StoreSettings.txt`
- `runtime\ThreadDump.txt`
- `runtime\ThreadList.txt`
- `runtime\TradingNetworks.txt`

The diagnostic utility is available beginning with Integration Server 6.5. For complete information about using the diagnostic utility, see the "*webMethods Integration Server Administrator's Guide*" or beginning with 8.0 SP1, see "*Administering webMethods Integration Server*".

Common Problems

Hanging or Unresponsive Integration Server

This is a common, general symptom with many different causes. All troubleshooting information is on the following page:

- [Integration Server Hang Troubleshooting](#)

Integration Server Crashes

An Integration Server crash refers to a sudden exit of the JVM process in which the Integration Server was running. All troubleshooting information is on the following page:

- [Integration Server Crash Troubleshooting](#)

JVM Problems

It is advisable to check the respective JVM vendor's pages for information about problems fixed in the latest minor (or 'underscore') release of the JVM.

IS Port Set to Deny by Default - "I locked myself out"

If you accidentally lock yourself out from the Integration Server, meaning you set the access mode on the only port to Deny by Default, you can always start the Integration Server with the `-port` command. Using `-port` will create a new port (that you specify) with Allow by Default access mode. This will allow you to change the settings back.

Example:

```
server.bat -port 5055
```

Unable to Locate Compiler

This issue occurs when Integration Server is unable to locate the compiler to compile any Java service. To resolve this issue, do the following:

1. Shut down Integration Server.
2. Open the `server.bat/server.sh` and edit the `JAVA_DIR` parameter to point to a JDK. Save the file.
3. Restart Integration Server.

This issue can also be caused by insufficient resources available to the JVM.

General Recommendations

Upgrade Database Driver to Latest Build of a Supported Version

Because problematic database drivers can cause the Integration Server to hang, Software AG recommends that you upgrade to the latest driver. To obtain the latest build of a database driver, do the following:

- If the driver is the OEM-version of Connect JDBC that Software AG provides, obtain the latest build of the driver from Global Support.
- If the driver is any other driver, obtain the latest build of the driver through your own support channels. (For example, by contacting Oracle for their JDBC drivers.)

Install the Latest JVM Patches

Check JVM patches to make sure that the latest patches are installed. JVM vendors typically fix hundreds of defects with each release, and many defects manifest themselves as hung processes. Software AG generally certifies support for a point release of a JVM (for example, 1.6.0), but Global Support often recommends upgrading to a specific underscore release (for example, 1.6.0_32) if it contains certain critical fixes from the JVM vendor.

Fixes

Software AG suggests reviewing the latest fixes. Fixes are solutions to known issues. You can find product fixes by product family, product, product version and operating system in the [Product Fixes area](#) on Empower.

Because fixes are cumulative, Readme files will provide the details about issues that are fixed by previous fixes as well. Readme files also contain the information about the fix requirements and installation instructions.

Software AG strongly recommends installing the latest service pack. New fixes will only be provided on top of the latest service pack. For example, assume that the current service pack of 8.0 is SP2 (that is, 8.0.2). In this case new fixes will only be delivered on top of 8.0.2; there will not be any new fixes created for 8.0.1.

Information to Provide When Opening a Support Incident

When you submit a Support Incident related to an Integration Server problem, please provide the following information to Software AG Global Support.

Clear Description of WHAT is the Problem

- Provide a short description of the problem you are facing.
 - For example, only specifying that you are seeing an error message is not enough. Apart from the error message, specify what is the side-effect, what is the problem?
- What is the business impact caused by the problem?

Environment Details About WHERE the Problem Is Occurring

- General information about WHERE ?
 - Where is the problem occurring? Dev/QA/Production ?
 - Are you able to reproduce the issue on any other environment?
- Obtain the following information from the Integration Server About page:
 - Operating system version
 - JVM version and JVM vendor

Description of WHEN the Problem Started/is Occurring

- When was the first time you started seeing the problem?
- How long were you running stable before the problem occurred? Or was the problem always there?
- How often is the problem occurring since it started? Can you recognize any pattern?

List of Installed Fixes

Obtain the list of installed fixes from the Integration Server About page.

Note! - Not all fixes and service packs display on this page after they have been installed. Also check the "updates" link on the bottom of the about-page.

- The best practice is to provide screen shots of:
 - IS About page
 - IS updates page
- Depending on the issue, it might also be helpful to provide screen shots from:
 - IS Extended settings
 - IS Clustering settings (if a cluster is involved)

Diagnostic Information

Because the nature of every problem can be different, so can the information required to troubleshoot and analyze the problem. However, the following information is typically requested for most Support Incidents submitted regarding Integration Server:

- **All Error Messages and/or Symptoms Witnessed**

Please provide complete error messages, including full stack traces (if any). There's a checkbox to see the stacktraces in the error-log screen. Please copy and paste the entire error message text. Because Software AG Global Support staff uses the exact text to search for previous reports of the same error text, avoid abbreviating information or introducing typos when copying the error messages.

- **All Integration Server Logs**

These logs are stored in either the `Integration Server/logs` directory or the database. The Integration Server logs include the following:

- audit log
- error log
- repository logs
- session logs
- stats log
- server log
 - In order to obtain useful server log entries, set the server debug level to 4 (Debug) or higher (Trace). Higher debug levels are preferable if your server load, performance requirements, and available filesystem space can accommodate them.
 - If you set the debug levels high (6 or above, called 'trace' in IS 7.1), selectively disable log facilities that generate many log messages but that are irrelevant to the problem. The log facilities we typically ask you to disable are the following:
 - 0038 HTTP Header
 - 0039 HTTP Request
 - 0040 HTTP Response
 - 0041 HTTP Cookie
 - 0049 Flow Operation
 - 0072 Reporter
 - 0066 Persistent Values Hash

- **All Integration Server Configuration Files**

This includes every file in the `Integration Server/config` directory. The actual file list depends on the Integration Server release but is typically the following:

- `acllist.cnf`
- `aclmap_sm.cnf`
- `aclread.cnf`
- `acls.cnf`
- `aclwrite.cnf`
- `broker_admin.cnf`
- `codegen.cnf`
- `dispatch.cnf`
- `eventcfg.bin`
- `invokemanager.cnf`
- `ISMJournalLogFilter.cnf`
- `ISMSvcExcptFilter.cnf`
- `jdbc/*` (and all subdirectories)
- `jobs.cnf`
- `PKIprofile.cnf`
- `port.cnf`
- `redir.cnf`
- `released_pkg.cnf`
- `remote.cnf`
- `repository2.cnf`
- `repository4.cnf`
- `server.cnf`
- `users.cnf`

- **A series of thread dumps** (This is only relevant when Integration Server appears to be "hanging/blocking")

- Make sure you also describe in what way the IS seems to be "hanging/blocking"
- Please provide a series of 5 thread dumps, taken at least 10 seconds apart. This helps Global Support to determine which threads are progressing and which threads are not progressing. Based on the nature of the problem, we may ask for different intervals.
- Instructions on obtaining Thread Dumps are described in the "[How To Obtain A Thread Dump](#)" section.

Integration Server - Performance

Note! Performance tuning of the Integration Server is typically not in the scope of services provided by Software AG Global Support.

However, we may be able to assist in troubleshooting performance issues that are isolated to a single component, are easily reproducible, and show results that are **clearly** outside the expected performance of a normal configuration.

About Performance Tuning

Many variables and factors can influence the performance of a webMethods Integration Server-based solution. A successful performance tuning effort requires on-site assistance and a detailed project plan. Resolving performance issues typically involves the following basic steps:

- Running a test case
- Identifying the performance bottleneck
- Removing the bottleneck
- Running the test case again

Collecting Performance Data

To collect system information, use the following:

- On Windows use the following commands:
 - Use Task Manager to get a first impression how the system behaves.
 - `Perfmon.exe` **Note:** limit yourself to the basic performance counters first.
 - To analyze problems concerning the state of the operating system use the utilities provided by sysinternals (now Microsoft) at <http://technet.microsoft.com/en-us/sysinternals/default.aspx>
- On Unix, use the following commands:
 - Use `ps` or `top` to get process information.
 - Use `vmstat` to obtain paging and CPU utilization information. You can use it to get disk utilization data for four devices (if you have many hard disks on your system, use `iostat` instead).
 - Use `iostat` to get disk I/O information.

File I/O

The following steps may be performed to reduce the amount of File I/O:

- Eliminate unnecessary audit logging.
Make sure the Integration Server's "watt.server.auditLog" parameter is set to "perSvc" and not to "brief" or "verbose". Example:

```
watt.server.auditLog=perSvc
```

- Check the audit settings for each service your solution uses and eliminate unnecessary audit points. For testing purposes, you might want to disable service logging entirely to gauge the impact of the current logging settings on performance. To do this, set the "watt.server.auditLog" parameter to "off" and then run your performance tests. Compare these results to the results produced when service logging is enabled, to determine whether logging is a significant bottleneck.
- Reduce the amount of debug logging.
Set the Level of Logging option in the Integration Server Administrator to a value that generates low amounts of logs. In most cases, FATAL, is sufficient. This Level of Logging records only fatal errors and critical messages.
- Avoid calls that result in disk I/O, such as the Java call

```
System.out.println
```

or services such as

```
pub.flow:debugLog
```


Garbage Collection

JVMs use Garbage Collection (GC) as a mechanism to reclaim unused memory. The garbage collection process is managed entirely by the JVM.

Garbage collection typically interrupts execution of all threads except the garbage collection helper threads. Therefore, frequent garbage collection can impact the performance of the Integration Server significantly. One simple way to lengthen the time between garbage collections is to increase the Java heap size. Note that this will also potentially lengthen the duration of each garbage collection pause, since the JVM will have to reclaim more memory.

To learn more about the garbage collection where your Integration Server is running, start the JVM with the `"-verbose:gc"` switch. This switch tells the JVM to log information relating to garbage collection. The logs contain information relating to frequency of garbage collection, length of time spent for each garbage collection, amount of memory reclaimed during each garbage collection, and so forth. For more information about the `"-verbose:gc"` switch, consult JVM documentation for the specific JVM in use.

Most JVM vendors provide a number of options that you can use to fine-tune the garbage collection process. Discussing different garbage collection options offered by different JVM vendors for different versions is beyond the scope of this document.

Heap Settings and Memory

Main Article: See [Memory Troubleshooting](#)

The amount of memory available to the Integration Server is determined by the size of the maximum heap you allocate to the JVM. The heap is the memory area in which the Java Virtual Machine (JVM) creates and stores objects. Because the heap is the only portion of memory that the Integration Server uses for data storage, the size of the heap (not the amount of memory installed on the machine) ultimately determines how many documents the Integration Server can actually process at one time.

- **Heaps that are too small**
A heap that is too small will produce out-of-memory errors. A small heap may also cause sluggish performance due to frequent pauses for garbage collection (reclamation of memory that is no longer in use).
- **Heaps that are too large**
Performance will also degrade significantly if the machine does not have enough physical memory to expand the heap to its configured maximum size and must then page portions of the heap to virtual storage (disk).

To determine how much of the heap your Integration Server uses, start the server and load only those packages that you expect to load under normal operating conditions. Immediately after the server loads, use the Integration Server Administrator to examine the **Available Memory** value on the "Settings > Resources" page. The "Available Memory" value will indicate how much of the heap is available for processing your documents. You can monitor this value as your solution executes to track its memory usage.

By default, the Integration Server is configured to allocate an initial heap size of 128MB, and configured to expand to a maximum heap size of 256MB. You can change the size of the heap memory in the Integration Server startup file:

- Windows: server.bat
- Unix/Linux: server.sh

To change the size of the heap, set the following Java parameters in the start-up file for the Integration Server:

- JAVA_MIN_MEM=<initialHeapSizeInMB>
- JAVA_MAX_MEM=<maximumHeapSizeInMB>

A common performance-tuning measure is to make the initial heap size (JAVA_MIN_MEM) equal to the maximum heap size (JAVA_MAX_MEM). Doing this prevents heap expansion and contraction, and can produce performance gains in some situations.

Some operating systems limit the amount of memory that a single process can use. For example, Windows does not allow a process to use more than 2 GB of memory. So, even if you equipped a Windows machine with 4 GB of memory, the Integration Server will never use more than 2 GB of it.

Some early versions of the JVM also restrict the maximum size of the heap. When tuning memory to improve performance, you need to be aware of any limitations that your existing operating system and/or JVM may impose. You may find that you need to switch operating systems and/or JVMs to achieve the heap size you need.

Memory Leaks

Main Article: See [Memory Troubleshooting](#)

Monitor Integration Server memory usage closely. If the Server's memory usage appears to grow geometrically over time, it is possible that a memory leak is consuming memory. Examine all your code (Flow and Java) for cases where objects are created but never destroyed.

Database

Main Article: See [Database Troubleshooting](#)

Performance problems that manifest themselves on the Integration Server may actually be caused by databases with which the Integration Server interacts. To determine whether a database issue is affecting performance, consider the following possibilities:

- **WmDB Connection Pooling:**

If you are using the WmDB package with Integration Server 4.6 or later, you can improve the performance of database-related services by enabling the connection pooling option on the Integration Server.

Note that by default, connection pooling is not enabled. To enable connection pooling, set the following option in the server.cnf file:

```
watt.server.db.connectionCache=server
```

Then code database services to close their database connections by calling `pub.db:close` when they are finished working with a database. Failure to do this will prevent the connection from being returned to the connection pool, and the connection will be lost. The pool will not grow to replace lost connections.

- **Database drivers:**

Change the database driver to determine if it is causing a performance problem. Upgrading the existing driver to a newer version or switching to a different driver vendor may improve performance.

- **Connection Pool configuration:**

Monitor the number of connections that the Integration Server maintains with the database during your performance testing. If the number of connections it maintains is equal to the maximum size of the Integration Server's connection pool, consider increasing the size of the connection pool.

Thread Pools

Integration Server maintains a variety of thread pools. The most important of these pools is the service thread pool. All client requests to invoke services are processed using thread(s) from the service thread pool. If Integration Server does not have enough threads in the service thread pool it will block the execution of a request. For instance, if 100 clients make a request to the Integration Server and the Integration Server has only 75 available threads in the thread pool, 75 of the requests will execute immediately. The remaining 25 requests will wait until threads become available in the service thread pool.

Note that in most cases, the service thread pool need not be equal to the number of clients. If clients invoke lightweight requests, then threads should become available quickly.

As a very general rule, the ideal number of threads falls in the range of 2-4 times the number of physical processors. In some cases, larger thread pools may increase throughput. However, larger thread pools do not always translate into higher throughput. When the Integration Server is permitted to execute a large number of threads concurrently, context switching can exert a noticeable effect on performance. The only way to determine the appropriate size of the thread pool is through testing. To identify the optimal pool size, gradually increase the number of threads in the pool and measure the performance under each size. (Note that every time you allocate more threads to the server thread pool, you should also increase the heap size, as every thread needs some amount memory for itself.)

Integration Server - Hang

The Integration Server, or services invoked on it, may at times appear to be hanging or unresponsive. This is a commonly reported problem symptom, and can have many different root causes. This section will help you identify the root cause of an Integration Server hang.

Common Symptoms

Any of the following symptoms may indicate that the Integration Server is in a hanging or unresponsive state:

- **Web Administrator Interface is Unresponsive**
This typically means that browser requests to the Web administrator interface connect but wait indefinitely for a server response that never arrives. The browser client eventually times out.
- **Developer Interface is Unresponsive**
This typically means that webMethods Developer requests to the Integration Server are able to connect but wait indefinitely for a response that never arrives.
- Service Invocation Hangs and Never Completes
- Server, Error, Stats Logs Are No Longer Being Written
- **JVM Shows 100% Heap Usage**
Determine whether the high CPU usage is being caused by the JVM process or some other process.
- Operating System Shows 100% (or higher) CPU Usage
- Operating System Shows 100% or more System Memory Usage

Note - For all of these situations to occur, the JVM process for Integration Server must still be alive. If the JVM process is no longer available, it may have crashed. Please refer to the [Integration Server Crash Troubleshooting](#) page for details.

Possible Causes and Resolutions

Apparent hanging or unresponsiveness of Integration Server may be caused by any of the following (in order of likelihood):

Excessive JVM Garbage Collection

Excessive JVM Garbage Collection activity occurs when Integration Server's JVM is unable to free enough memory during garbage collection cycles. The JVM continually blocks the Integration Server threads from continuing while it is collecting garbage, preventing the Integration Server from progressing. This cause also typically evokes the symptom of near-100% CPU usage, which gives the appearance of a hanging system even though the system is not hung but just progressing very slowly. Such memory exhaustion is sometimes caused by excessive memory usage by your implementation, not by any product defects.

- Shut down and restart Integration Server.
This is only an interim solution. If the root cause is not resolved, the problem will likely recur.
 - Shut down the Integration Server via regular means if possible.
 - After issuing the shutdown command, check for the existence of the Integration Server's Java process and wait for it to terminate. You can also check the server log for the 'Shutting down server' message.
 - If the Integration Server does not finish shutting down, you may need to terminate the process via TaskManager in Windows or using the kill command in Unix.
- Reduce the amount of memory used.
This is the preferred long-term resolution. By changing the way documents are processed, in many cases the amount of memory needed can be reduced. More advanced and detailed memory tuning services are offered by Software AG Global Consulting Services.

Excessive CPU Usage

Excessive CPU usage (for example, near 100%) might be caused by the Integration Server process or by other processes not related to the Integration Server JVM. It is therefore important to first identify which processes are the primary contributors to the high CPU usage. Please refer to the [CPU Troubleshooting](#) section for details.

Bottleneck in Audit Subsystem

An incorrectly configured/tuned asynchronous auditing mechanism can severely slow down Integration Server execution. For information about tuning the Audit Subsystem, please refer to the "*webMethods Logging Guide*."

Exhausted Integration Server Service Thread Pool

You should see a warning and error message in the server logs indicating this problem. Either increase the amount of available service threads (if the JVM has enough free memory to accommodate more threads), or change the integration logic/configuration to ensure less threads are needed.

Unresponsive Database

Many Integration Server components depend on a functioning database. The Audit Subsystem is one example. It is common for the Audit Subsystem to be heavily used and configured to write to the database. If queries to the database fail to complete, or take a very long time to complete, any service that is configured for auditing will also fail to complete. This in turn can hang the entire Audit Subsystem, which can result in a sluggish or hung Integration Server. There may also be custom services that hold or lock resources which are hanging on database calls.

Unresponsive Backend System

The Integration Server can hang if backend systems are not responding properly. This type of hang can occur if the thread pool is completely consumed by services that require replies from the unresponsive system. You can usually diagnose this type of hang from a thread dump.

To prevent the Integration Server from hanging indefinitely when a backend system stops responding, set the `watt.net.timeout` Integration Server property. `watt.net.timeout` is used for outbound connections (that is, the request goes out from Integration Server).

Note - In recent versions of the Integration Server, this setting can be overwritten with an optional input of the `pub.client:http` service. For additional information about these properties, see the *"webMethods Integration Server Administrator's Guide"*.

Hanging Code Inside (Database) Driver

The Integration Server can hand over control to many third party (Java) libraries, for example a Database Driver. A problem in such a third party library can block the entire Integration Server. Thread dumps are helpful in identifying this situation.

Java Thread/Monitor Deadlock

The JVM and several layers of the Integration Server application try to avoid a deadlock situation. Generally, the Integration Server cannot recover from a true thread deadlock. You must terminate the JVM to resolve a deadlock condition, because Java provides no safe mechanism for terminating individual threads. Obtain thread dumps to determine whether a thread deadlock caused the Integration Server to hang and to discover what caused the deadlock. Software AG Global Support can assist in analyzing the thread dumps for evidence of a deadlock.

Blocked Finalizer Thread

The "Finalizer" is a special thread used by the JVM during garbage collection. It invokes "finalizer" methods on objects that are scheduled to be garbage collected. Some finalizer methods (for example, in third-party libraries) can potentially hang, which can result in a hanging Finalizer thread--especially when database drivers attempt to close connections. The Integration Server's critical systems rely on correct operation of the Finalizer thread that, if blocked, will result in a Cascading Java Monitor problem and the appearance of an unresponsive or hanging Integration Server.

Check for a Hanging Finalizer Thread

To check for a hanging Finalizer thread, look for the thread titled "Finalizer" in the thread dump.

A correctly-functioning, **non-hanging** Finalizer thread appears in the dump as follows (the thread status "Object.wait()" is typical of a normal Finalizer thread):

```
"Finalizer" daemon prio=9 tid=0x009ce4d0 nid=0xae4 in Object.wait() [2c6f000..2c6fd8c]
at java.lang.Object.wait(Native Method)
at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:111)
locked <0x113c02e0> (a java.lang.ref.ReferenceQueue$Lock)
at java.lang.ref.ReferenceQueue.remove(ReferenceQueue.java:127)
at java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:159)
```

A **hanging** Finalizer thread may appear as follows (the thread status "waiting for monitor entry" is typical of a hanging Finalizer thread and indicates that the thread is waiting for a monitor held by another thread):

```
"Finalizer" daemon prio=9 tid=0x0095cd10 nid=0x2ac waiting for monitor entry [6470f000..6470fd88]
at com.wm.dd.jdbc.base.BasePreparedStatement.close(Unknown Source)
waiting to lock <0x257bb5f0> (a com.wm.dd.jdbc.oracle.OracleConnection)
at com.wm.dd.jdbc.base.BaseStatement.finalize(Unknown Source)
at java.lang.ref.Finalizer.invokeFinalizeMethod(Native Method)
at java.lang.ref.Finalizer.runFinalizer(Finalizer.java:83)
at java.lang.ref.Finalizer.access$100(Finalizer.java:14)
at java.lang.ref.Finalizer$FinalizerThread.run(Finalizer.java:160)
```

In this example, the Finalizer thread is waiting for the monitor <0x257bb5f0>. The root cause of the problem is not the Finalizer thread itself, but rather the thread that is holding that monitor and not releasing it. In the above thread, the trail led to a thread with the following appearance (notice that it has locked the monitor <0x257bb5f0>).

```
"HTTP Handler 10.80.11.237" prio=5 tid=0x00a06340 nid=0xc30 runnable [6595f000..6595fd88]
at com.wm.dd.jdbc.oracle.net8.OracleDataProvider.readBytes(Unknown Source)
at com.wm.dd.jdbc.oracle.net8.TTIDataPacket.readCLRInBuf(Unknown Source)
at com.wm.dd.jdbc.oracle.net8.TTIRXDDataPacket.readRXDDDataInBuffer(Unknown Source)
at com.wm.dd.jdbc.oracle.OracleImplBlob.readData(Unknown Source)
at com.wm.dd.jdbc.base.BaseImplBlobChunked.readData(Unknown Source)
at com.wm.dd.jdbc.base.BaseBlobInputStream.read(Unknown Source)
at com.wm.dd.jdbc.base.BaseInputStreamWrapper.read(Unknown Source)
locked <0x257bb5f0> (a com.wm.dd.jdbc.oracle.OracleConnection)
at java.io.BufferedInputStream.read1(BufferedInputStream.java:220)
at java.io.BufferedInputStream.read(BufferedInputStream.java:277)
locked <0x2669dfd0> (a java.io.BufferedInputStream)
at com.wm.app.tn.db.BDSingleDocOperations.getBytesFromStream(BDSingleDocOperations.java:401)
at com.wm.app.tn.db.BDSingleDocOperations.get(BDSingleDocOperations.java:288)
at com.wm.app.tn.db.BDSingleDocOperations.get(BDSingleDocOperations.java:86)
at com.wm.app.tn.db.BizDocStore.getDocument(BizDocStore.java:293)
at wm.tn.doc.view(doc.java:104)
at sun.reflect.GeneratedMethodAccessor114.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:324)
at com.wm.app.b2b.server.JavaService.baseInvoke(JavaService.java:287)
at com.wm.app.b2b.server.ServiceManager.invoke(ServiceManager.java:347)
at com.wm.app.b2b.server.ServiceManager.invoke(ServiceManager.java:119)
at com.wm.app.b2b.server.HTTPInvokeHandler._process(HTTPInvokeHandler.java:138)
at com.wm.app.b2b.server.InvokeHandler.process(InvokeHandler.java:113)
at com.wm.app.b2b.server.Dispatch.run(Dispatch.java:204)
at com.wm.util.pool.PooledThread.run(PooledThread.java:105)
locked <0x164d74b0> (a com.wm.app.b2b.server.TMPooledThread)
at java.lang.Thread.run(Thread.java:534)
```

The corrective action depends on what is causing the thread to hang. In this case, the cause seems related to the database.

General Recommendations

General Recommendations regarding the Integration Server are described in the [General Recommendations](#) section.

Information to Gather When Opening a Support Incident

To troubleshoot an Integration Server hang, obtain the following information:

- Provide the information listed on the **Integration Server General Information** page.
- Obtain a series of [Integration Server thread dumps](#) while the hang is happening and before Integration Server is restarted. Please make sure to capture thread dumps 30 seconds to 1 minute apart. It is important to capture multiple thread dumps as it is helpful in comparing the state of the threads as time progresses.
This information provides important insight into what the server was doing when the hang occurred.
- Invoke `wm.server:ping` on the Integration Server from a Web browser:
 - Open a Web browser.
 - Invoke the URL `http://<hostname>:<port>/invoke/wm.server/ping`, where "hostname" and "port" refer to the host and port where Integration Server is running.
 - Monitor the Web browser to determine if the browser hangs or gets a response from the Integration Server.
- **Provide details about when the problem was first witnessed.**
Specify the exact time at which the Integration Server's unresponsiveness became apparent, so that the logs written near this time may be inspected.
- **Obtain garbage collection (GC) logs.**
Enable logging of garbage collection information with the appropriate JVM options in `server.bat/server.sh`. For example `"-verbose:gc"`.

Integration Server - Crash

An **Integration Server crash** refers to a sudden exit of the Java Virtual Machine (JVM) process in which the Integration Server was running. The Integration Server itself is written in Java and its code - and errors contained within - are therefore not capable of directly causing a crash. Instead, these crashes are usually caused by JVM defects, ⁽¹⁾ or code executing inside native libraries. ⁽²⁾

Important Note! - *Integration Server crashes are often caused by JVM defects or third-party native libraries, both of which are non-Software AG software. To obtain support for this non-Software AG software, the customer must engage the support group of the company that supports that software.*

Common Symptoms

An Integration Server crash refers to a sudden exit of the JVM process in which the Integration Server was running. Therefore, in a crash, the first obvious symptom would be the disappearing of the Java process.

In addition, when the JVM crashes, it generally creates a "hs_err_pid<pid>.log" file.

Possible Causes and Resolutions

The JVM provides significant insulation from the operating system. Java's lack of direct memory manipulation methods and robust exception handling mechanisms prevent most types of process crashes. Because of this, few conditions cause the Integration Server process to crash. Following are known conditions that cause the Integration Server process to crash.

JVM Bugs

JVM defects do exist, and they can cause unexpected JVM crashes. JVM vendors typically fix hundreds of defects with each release, and many defects manifest themselves as process crashes. JVM bug lists are typically available online on the JVM vendors' Web sites. Software AG generally recommends upgrading to the latest minor ("_") release of the JVM supported by the Integration Server. For example, if you are on Integration Server 7.1.2, JVM 1.5 is supported; therefore, you would upgrade to (for example) 1.5.0_22. **Note** - JVM 1.5.0_22 was the last build to be released before the 1.5 version went into "End Of Life" state. Contact Software AG Global Support if a known defect is solved in a newer JVM build.

Often, JVM bugs are in the just-in-time (JIT) compiler. To troubleshoot JVM crashes, consider disabling JIT. However, because this action may severely affect performance, test this first. To disable the JIT, set the System property in `java.compiler` to **NONE**. (The **-nojit** option is no longer valid.)

Native Code

Java provides a way to execute native, or non-Java, code (for example, C++ code) from within the JVM. Defects in native code can crash the JVM process. Therefore, if a service calls non-Java code from the Integration Server, be aware that defects in that code can cause the Integration Server to crash.

Native code that might cause the Integration Server to crash includes:

- Custom native code that is invoked directly from the Integration Server (custom native services developed by users)
- Native code libraries associated with an adapter running on the Integration Server (for example, SAP Adapter's jCo libraries, or MQSeries)
- Native code libraries associated with type II database drivers; if you are using a type II JDBC driver, try switching to a type IV JDBC driver, which is a pure Java-based, or thin, driver.

Note that when native code causes a crash, the entire Integration Server crashes. This is an inherent limitation of Java programs.

Processing Large IDOCs using SAP Libraries

See *"Integration Server process crashes when processing large IDOCs"* on the [SAP Adapter Troubleshooting page](#) for details.

Logging Out of Windows

With certain JVM versions, a JVM process running on Windows terminates when the user who started the process logs off. To avoid this problem, use the `-Xrs` JVM switch (even when running the Integration Server as a Windows service). This switch instructs the JVM to ignore certain external signals, including a logoff signal. If the Integration Server is running as a Windows Service and if the JVM is shutting down abruptly, make sure that the `server.bat` file contains the `-Xrs` switch as an argument to the Integration Server JVM. In the event of the server being shutdown due to the logout, the logs would display the `"[ISS.0014.0005I] Shutting down server. JVM Shutdown Hook."`

Lack of Memory for the Native Heap

Java allocates some of its process memory to its native heap. This heap is used by native code (if any), loading classes, and certain types of Java objects. Generally, JVM allocates the Java heap as specified by the `-mx` and `-ms` parameters and uses the leftover memory for its native heap. If the JVM is unable to allocate the memory it needs, it crashes. If you are using native code (see above), consider reducing the Java heap size so that native heap can obtain the quantity of memory it needs, as follows:

- Reduce the value of `-mx` in the `server.bat` or `server.sh` file.
- Reduce the stack size by setting the value `-Xss256K` (to allocate 256 KB of thread stack) or `-Xss128K` (to allocate 128 KB of thread stack). For more information about this parameter, see the Java documentation.

Others

Other factors that may cause an Integration Server to crash include the following:

- **Operating System Defects:** Ensure the operating system patch levels are up to date.
- **File Descriptors:** An insufficient number of file descriptors can cause the Integration Server to crash on UNIX platforms. Set the file descriptor limit to a large number such as 2048 or 4096.
- **Disk Space:** Ensure that the platform is not running out of disk space.

General Recommendations

The Software AG webMethods Integration Server is written in Java and runs inside a Java Virtual Machine (JVM). The JVM is provided by another vendor (typically IBM or Oracle) and Software AG does not have access to the source code of the JVM. Troubleshooting JVM issues often requires joint coordination with the support team of the JVM vendor. In Software AG's experience, the quickest and most effective way to achieve this is for the customer to open a support ticket with the JVM vendor, inform Software AG Global Support of the ticket number, and coordinate joint action between the two vendors' support engineers.

See further recommendations in the [Integration Server Troubleshooting section](#)

Information to Gather When Opening a Support Incident

Most crashes occur due to a fault in the JVM or in the native (JNI) code being run in the Java process. Therefore, if the application does not include any JNI code and does not use any third-party packages that have JNI code (for example, JDBC application drivers), the fault must be in the JVM and should be reported to the JVM vendor through the normal process.

You can report the problem to Software AG Global Support to help isolate the issue. When you open a Support Incident related to Integration Server crashes, please provide the following information.

Environment details

Screenshots of the Integration Server About Page (includes OS Vendor / Version, JVM Vendor / Version, 32-bit / 64bit)

"hs_err_pid<pid>.log" file

Sun and HP JVM's use this file to record information about a crash. "hs" stands for Hotspot, "err" for Error and "pid" indicates the Process ID under which the JVM was running. This file can be very helpful for determining the cause of the crash. It contains information about the active threads and the call stack at the time of the crash.

The IBM JVM records similar information in a "java core" core file in the event of a crash.

nohup.out or stdout of the JVM Process

When the JVM process crashes, it typically writes Java core containing more information about the thread the JVM was using when the process crashed. It also gives information such as the min/max memory available during the crash. This information helps pinpoint the cause of the crash.

Verbose JVM JNI Output

The JVM can log verbose JNI information. These logs can help you determine, for example, if a JNI call was entered and never exited, an indication that the crash may have been caused by code inside that JNI call.

To enable verbose JNI logging information, add the following parameter to the Java command in server.bat or server.sh:

```
-verbose:jni
```

UNIX truss Output

The UNIX truss command shows the low-level system operations executed by the JVM, including the following helpful information:

- **What files are being accessed on the filesystem**
This might help indicate the point during the startup process at which the Integration Server encountered problems.
- **The JVM process exit code**
The exit or return code might help indicate the reason the JVM process exited.

To use the truss command, either add it to the `server.sh` file and then start the Integration Server or attach it to a running Integration Server instance. For more information about the syntax of this command, see the man pages.

UNIX Core Dump

Do not send UNIX core dumps to Software AG Global Support. Software AG cannot analyze JVM core dumps. Instead, open a ticket with the JVM vendor and send them the core dump.

Windows Dr. Watson Logs

If a program error occurs, [Dr. Watson](#) may start automatically. You can also manually start Dr. Watson by performing one of the following:

- Click Start > Run and type `drwtsn32`.
- From a command prompt, change to the root directory and type **`drwtsn32`**. On the subsequent dialog box, select the path where the log file and the core dump files will be saved. Also select log options as desired.

If you have been using a program other than Dr. Watson as your default debugging tool and you want to use Dr. Watson instead, at the command prompt type the command **`drwtsn32 -i`** to start Dr. Watson. The **`-i`** option makes the necessary changes to the registry.

References

1. Sun JVM Bug ID 4697804: VM should minimize need for calls to `vm_exit_out_of_memory()`

[Sun JVM Bug ID 4697804: VM should minimize need for calls to `vm_exit_out_of_memory\(\)`](#)

2. SAP OSS Note 634689 (Version 4): Central Note for Memory Issues, SAP J2EE Engine 6.20

According to Sun, the system state after an `OutOfMemoryError` is unpredictable and the system should be restarted.

Memory problems in native (JNI) part

Some Java packages use methods implemented in C/C++ via Java Native Interface (JNI)--for example, Java Connector (JCO). The native coding can also allocate memory via `malloc()`. In some cases JCO catches the allocation errors in the C-coding and throws an "out of memory" error ("Out of memory, last request was %d bytes"). If you see such an error, you can be sure it comes from the JNI part of JCO.

Increasing the **-Xmx** value (that is, the memory reserved for Java) decreases the amount of memory available to JNI. In this case, decrease the value of **-Xmx**. Unhandled allocation failures in the native coding can also cause process crashes with "ACCESS_VIOLATION".

In such a case, a HotSpot Error usually occurs. Check the server folder for the presence of `hs_err*`, `javacore*`, `java_core*`, and core files.

Integration Server - Startup

Common Symptoms

An Integration Server startup failure is a common symptom with many potential causes. The startup sequence may fail at various points and may exhibit the following symptoms:

- Integration Server exits before the startup sequence logging ever begins.
- Integration Server exits at some point during the main startup sequence.
- Integration Server hangs at some point during the startup sequence.

Possible Causes and Resolutions

The typical reasons an Integration Server experiences problems during startup are (in order of prevalence):

- Problems with files in the server "IntegrationServer/config/" directory
- Problems with files in the "IntegrationServer/DocumentStore/" directory
- Problems with files in the "IntegrationServer/audit/data/" directory
- Problems connecting to an external repository server (before 7.x)
- No Integration Server ports can start because of a lack of port resources or conflicts with an OS port or another Integration Server instance

Changes to the server.bat File Are Not Reflected After Restarting Integration Server as a Service

If you modify the server.bat file (for example, change memory settings) and you are running Integration Server as a service on Windows, do one of the following to ensure that the changes take effect:

- Run Integration Server from the command prompt with the -service option. This updates the Windows registry to reflect the changes.

```
server.bat -service <servicename>
```

- Unregister Integration Server as a service from the "IntegrationServer/support/win32" directory by running the command "InstallSvc.bat unreg". Then, register the service using the command InstallSvc.bat reg. This process updates the Windows registry to reflect the changes from the modified server.bat file.
- Update the Windows registry directly by editing the registry key "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\wmIS\WebmServiceInfo", making the required changes to the JVM settings. This is not the preferred method.

Integration Server Exits Immediately Before or During Startup Sequence

Problems with the contents of the files in the "IntegrationServer/audit" directory can cause Integration Server to exit immediately before or during the startup sequence. The AuditLogManager exception indicates this situation as follows:

```
2005-11-14 09:30:58 CET [ISS.0095.0001C] AuditLogManager Initialization Exception:
com.wm.app.audit.AuditException: [BAA.0002.0000]
Wrapped Exception: com.wm.app.repov4.RepositoryException: [BAR.0002.0000]
Wrapped Exception: com.wm.data.DataException: [BAC.0002.0000]
Wrapped Exception: com.wm.io.codec.CodecException: [BAC.0005.0004] CodecException: Unknown Object
Type 28
com.wm.io.codec.CodecException: [BAC.0005.0004] CodecException: Unknown Object Type 28
at com.wm.util.codec.BinaryCodec.unknownDecodeType(BinaryCodec.java:463)
at com.wm.util.codec.BinaryCodec._decode(BinaryCodec.java(Compiled Code))
.....
```

To resolve this error, try one of the following:

- Rename the audit directory in the IntegrationServer directory. Integration Server creates a new, empty audit directory during the next successful startup sequence. **Note:** This solution deletes all of the information in the Document Store; therefore, back up the files in the Document Store before deleting or renaming the audit directory.
- Rename the "IntegrationServer/DocumentStore" directory. Integration Server creates a new, empty directory during the next successful startup sequence. **Note:** This solution deletes all of the information in the Document Store; therefore, back up the files in the Document Store before deleting or renaming the DocumentStore directory.
- Rename the "IntegrationServer/WmRepository4" Directory. Integration Server creates a new, empty directory during the next successful startup sequence. **Note:** This solution deletes all of the information in the Document Store; therefore, back up the files in the Document Store before deleting or renaming the WmRepository4 directory.

For more crash-related troubleshooting, see [Integration Server Crash Troubleshooting](#).

Integration Server Hangs During Startup

Integration Server can fail at various points during startup. To determine the point at which it fails, examine the server log.

Hangs After the Line [SCC.0126.0001E] SCC ConnectionManager Pool

The last lines in the server log are:

```
...
[ISS.0028.0012C] WmJDBCAdapter: Startup
[SCC.0126.0001E] SCC ConnectionManager Pool
```

The startup sequence never reaches the "ConnectionManager Pool Started" status. In this case, Integration Server may have stopped loading after it tried to load the JDBC Adapter package because a database it was trying to connect to was not available.

To resolve this issue, first temporarily disable the JDBC Adapter package by editing the `manifest.v3` file and changing `enabled=yes` to `enabled=no`. Then, start Integration Server and re-enable the JDBC Adapter from the `Packages > Management` section on the Integration Server Administrator.

Hangs for a Couple of Minutes, Prints "Skipped Compiling {java/lang/Thread}{sleep}" On the Console, and Continues Startup Sequence

The console message is the result of a directive to the IBM JVM to skip JIT optimization of the `java.lang.Thread.sleep()` method. It is not a directive for the Integration Server startup sequence to sleep.

Ensure Ports Used by Integration Server Are Not Already in Use

You can use the `netstat` command to determine which ports are being used.

Example:

```
C:\>netstat -a
Active Connections
Proto Local Address Foreign Address State
TCP ThisHost:5555 ThisHost.domain:0 LISTENING
```

General Recommendations

General Recommendations regarding the Integration Server are described in the [General Recommendations](#) section.

Information to Gather When Opening a Support Incident

When you submit a Support Incident for Integration Server startup problems, please provide the following information to Software AG Global Support.

Server Log

The `server.log` indicates the sequence and progress (or lack thereof) of the startup sequence of Integration Server.

Thread Dumps

If the Integration Server startup is 'hanging' it is helpful to obtain a series of thread dumps, at the point where it hangs.

- Instructions on obtaining thread dumps can be found on the "[How to obtain a JVM Thread Dump](#)" page.

Verbose JVM Class Loading Output

Verbose JVM class loading output displays the classes that are loaded by the JVM. This method is crude, but it provides insight into how much progress was made by the JVM before problems were encountered.

To enable verbose JVM class loading output, add the following line to the JVM command:

- `-verbose:class`

UNIX truss Output

The UNIX **truss** command shows the low-level system operations executed by the JVM, including:

- What files are being accessed on the filesystem
This information might help indicate the point during the startup process at which Integration Server encountered problems.
- The JVM process exit code (if the process is exiting for some unknown reason)
The exit or return code might help indicate the reason the JVM process exited.

To use the truss command, either add the command to the `server.sh` file and then start Integration Server or attach it to a running Integration Server instance. For information about the syntax of this command, see the man pages.

Integration Server - Shutdown

Common Symptoms

An Integration Server shutdown failure has many potential causes. The shutdown sequence may fail at various points and may exhibit the following symptoms:

- The Integration Server shutdown sequence hangs indefinitely.
- The Integration Server shutdown sequence encounters significant delays before eventual completion.

Possible Causes and Resolutions

- Shutdown sequence hangs on a trigger

Set the following extended property to true:

```
watt.server.trigger.interruptRetryOnShutdown=true
```

`watt.server.trigger.interruptRetryOnShutdown:`

Specifies whether a request to shut down the Integration Server interrupts the retry process for a trigger service. If this parameter is set to false, the Integration Server waits for the maximum retry attempts to be made before shutting down. The Integration Server will also shut down if the trigger service executes successfully during a retry attempt. If this parameter is set to true, the Integration Server waits for the current service retry to complete. If the trigger service needs to be retried again (the service ends because of an `ISRuntimeException`), the Integration Server stops the retry process and shuts down. Upon restart, the transport (the Broker or, for a local publish, the transient store) redelivers the document to the trigger for processing. The default is false.

General Recommendations

General Recommendations regarding the Integration Server are described in the [General Recommendations](#) section.

Information to Gather When Opening a Support Incident

As with all hang situations, obtain a series of Integration Server thread dumps while the hang is happening and before Integration Server is restarted.

This information provides important insight into what the server was doing when the hang occurred.

- Instructions on obtaining thread dumps can be found on the "[How to obtain a JVM Thread Dump](#)" page.

Integration Server - ART

Common Problems / Typical Causes

In any situation where an adapter is having problems with connectivity to a resource, the ART should be looked at. The ART is actually responsible for creating the connection to the resource, maintaining the pool of connections and reconnecting in case a connection is lost.

Typical causes could be related to the network configuration (for example, switches or firewalls). However, it is always advisable to make sure that the latest WmART fix has been installed. These are part of the Integration Server Core fixes.

Connections to Resource Not Reestablished After Connection Is Broken

A connection in the connection pool will not be refreshed immediately after connections are broken. Connections are only refreshed under the following conditions:

Adapter Services:

- Invocation of an adapter service results in a connection error.

Polling Notifications:

- Execution of a polling notification on its scheduled interval encounters a connection error.

Connection Statistics for an ART Connection

To obtain statistics for an ART connection or resource, use the following service:

- (WmART) `pub.art.connection:getConnectionStatistics`

Information to Provide When Opening a Support Incident

When you submit a Support Incident for an Adapter Runtime issue, please provide the following information to Software AG Global Support:

- Integration Server's Server log.
- Integration Server's Error log with expanded stack traces.
- Screen shots of the Integration Server's "About" and "Updates" pages to show the Integration Server version and patch level.
- Screen shot of the Package Details page of the WmART package.
 - In Integration Server Administrator, go to "Packages" - "Management" - Click on the link for "WmART". The bottom of this page mentions the "Patch History".
- Screen shot of the Package Details page of the adapter you're using.
- Screen shot of the Connection configuration settings for the connection that has the problem.
 - In Integration Server Administrator, go to "Packages" - "Management" - Click on the "Home" icon for the adapter in question.
- Connection run time statistics obtained from the built-in service `pub.art.connection:getConnectionStatistics`.
- Application package (if the issue is related to transaction handling).
- [Thread dumps](#) (if the issue pertains to performance or a non-responsive server or thread).

Integration Server - Cluster

Common Symptoms

Any of the following symptoms may indicate that there is an Integration Server cluster related issue:

- Cluster Hosts section of the Cluster settings page does not contain all nodes that are in the cluster.
- Tangosol Coherence related exceptions or errors are generated during Integration Server startup.
- Guaranteed delivery services are not working as expected in a cluster.
- Cluster-enabled scheduled jobs are running on more than one node at a given time.

Possible Causes and Resolutions

Cluster Nodes Unable to Exchange Multicast Packets Over the Network

In a clustered configuration, the session state for clients connected to all servers in the cluster is stored in the shared distributed cache created by Oracle Coherence. The cache allows transactions in a conversation to be continued on other servers in the cluster.

For a valid cluster configuration, ensure that the values for "Cluster Name", "Discovery Address" and "Discovery Port" are the same on all the cluster nodes (Integration Server). Also ensure that the address specified for "Discovery Address" is open on your network, and that the port specified for "Discovery Port" is available.

To check if an Integration Server that is hosted on a different server on your network can participate in a cluster, you can perform a multicast test on all the servers.

Instructions for Performing a Multicast Test:

Perform the multicast test between two machines that host Integration Servers forming the cluster. This test will verify whether the Integration Servers running on these two physical machines will be able to exchange multicast packets over the network. The ability to transmit and receive multicast packets is integral for Tangosol clustering.

Please execute the following test utility for all servers participating in the cluster. It is not necessary for the Integration Servers to be up and running.

On server1, hosting one Integration Server cluster node:

1. cd to <webm_install>\IntegrationServer.

2. Run the following command:

```
java -classpath "<full_path_for_coherence.jar>" com.tangosol.net.MulticastTest -group
<Discovery_address:Discovery_port>
```

For example:

```
java -classpath "C:\webMethods712\common\lib\ext\coherence.jar"
com.tangosol.net.MutlticastTest -group 224.12.12.19:24988
```

You can use the same values for Discovery_address and Discovery_port that are used for the Integration Server cluster configuration.

3. The multicast test starts on server1, and will show the packets sent and received. Initially the multicast test will receive packets only from itself.

4. Repeat the above steps for server2, hosting a second Integration Server cluster node. When the multicast test starts running on this server as well, it should show that packets are received from server1 (ipaddress), as well.

Server Clocks are Not Synchronized

Ensure that the time on all the installations of Integration Server is synchronized. (You can use a tool such as Time Server.) Clock synchronization is particularly important for scheduled tasks that are "cluster aware."

Differences Between the Cluster Nodes

Ensure that all the Integration Servers participating in the cluster are identical to each other in terms of `server.cnf` settings, fixes, license keys, JVM parameters, and operating system settings as well as the custom application code.

Clustering Not Enabled On All the Cluster Nodes

Ensure that clustering has been enabled on all Integration Servers, so that all the Integration Server nodes can join the shared distributed cache to become part of cluster.

Access Mode and IP Access Configuration

Ensure the Access Mode and IP Access settings for the ports are set up correctly.

ISInternal JDBC Pool Not Pointing to the Same Database Schema

Information about scheduled jobs, jobs tracked by guaranteed delivery, and XREF data is stored in an external database, identified by the `ISInternal` functional alias. It is very important to make sure that all Integration Servers that are part of a cluster point to the same "ISInternal" database schema.

General Recommendations

The following configuration files in `IntegrationServer/config/Caching` should not be changed manually:

- `Coherence-cache-config.xml`
- `Tangosol-coherence.xml`

See further general recommendations in the [Integration Server Troubleshooting](#) section

Information to Gather When Opening a Support Incident

To troubleshoot an Integration Server cluster related issue, obtain the following information:

- Provide the information listed on the [Integration Server General Information](#) page.
- For each Integration Server node in the cluster, provide:
 - A screen shot of the Integration Server Administrator "About" page.
 - A screen shot of the "Settings > Clustering" page from the Integration Server Administrator.
 - Integration Server log with "Cluster Manager (033)" facility set to "Trace" logging level.
 - Integration Server console output, if available; on UNIX, this is usually the `nohup` file.
 - Integration Server error log; if error log is in the database, include the stack trace when extracting the log output.
 - Output of the command

```
java -classpath "<full_path_for_tangosol.jar>"
com.tangosol.license.ProcessorInfo
```

For example:

```
java -classpath "C:\webMethods712\common\lib\ext\tangosol.jar"
"com.tangosol.license.ProcessorInfo"
```

Integration Server - Pub/Sub

Introduction

Many integrations make use of the functionality for publish and subscribe offered by webMethods Integration Server. Usually a webMethods Broker will be used for storage and distribution of the documents. Because publish and subscribe functionality involves both a sender and a receiver, problems can reside at either end.

This section describes common symptoms, causes, and resolutions of problems that can be encountered in Integration Server when using publish and subscribe.

Common Symptoms

The following symptoms can be observed when dealing with publish/subscribe issues.

Publishing Documents to Broker

- **Publishing fails with an error**
An error can be displayed in webMethods Developer, My webMethods Server screens, or Integration Server log files indicating that a publish operation has failed. The published document will not be sent to the Broker.
- **Publishing succeeds but the document does not reach the Broker**
No errors are observed when publishing a document, but the document is not received by the Broker.
- **Publishing does not finish**
Integration Server starts to publish a document, but does not finish the operation. The service does not return and the Java thread remains active.

Subscribing to Documents from Broker

- **An error occurs while receiving a document**
An error can be displayed in the Integration Server log files, indicating that there is a problem with the trigger. The document is not received from the Broker.
- **Documents appear to be lost**
Even though the storage type is set to Guaranteed, the document is not in a Broker queue, nor has it been processed by Integration Server.
- A request is published, but the reply is not handled
- **The trigger service is not retried upon failure**
There was a failure in the service processing a received document, but the trigger already discarded the document.

Possible Causes and Resolutions

Publishing Documents to Broker

Broker does not have enough storage to process the request

If the message "Broker does not have enough storage to process the request" displays while you are publishing documents to the Broker, do the following:

- Check whether you have enough storage space in Broker. To increase the queue storage space in the Broker, use the `server_config` command. For information about using this command, see the *"webMethods Broker Administrator's Guide"*.
- Ensure that you have applied the latest Broker service pack.
- Check the maximum transaction size of the Broker Server's storage session. You may see the message described above if you are publishing a document or a set of documents that is larger than the maximum transaction size.

Documents are not saved in Outbound Document Store when Integration Server is not connected to the Broker or Broker is down

Possible causes for this issue include the following:

- The outbound document store is full. Check the configured size and how many documents are currently in the document store with the Integration Server Administrator, Resources, Store Settings.
- The `watt.server.control.maxPersist` property in `server.cnf` is set to 0. With this setting, the outbound document store is disabled and no documents are stored there. If the Broker is down for an extended period in a high-volume environment, the entire service thread pool could potentially block while waiting for the Broker to come back up.

Subscribing to Documents from Broker

Trigger execution does not work as expected

If the trigger is not executing as expected, do the following:

- Make sure that the document firing the trigger is in the correct format and contains values in all mandatory fields.
- Determine whether the problem is with the trigger or with another service. Sometimes, the main service called by a trigger may invoke many other services. For example, the service may map the incoming document, perform some validation, and then call an adapter service to insert a row into a database. If the row is not inserted in the database, you may suspect that the trigger is not working correctly. To help isolate the problem, write a debug statement at the beginning of the invoked service to determine whether the problem is with the trigger or in the downstream services.

- Check the Time to Live property of the published document. This property defines the time the document will remain in the Broker. You can set this property to Never Discard to make sure that the documents are not discarded from the Broker because of a timeout issue.

Guaranteed documents appear to be lost

The Broker's guaranteed storage mechanism is robust and uses a two-phase commit to store the guaranteed documents. You can expect "at least once" delivery of every guaranteed document. Documents may be delivered more than once if you do not have a duplicate checking mechanism in place. However, you should never lose a document.

The most common errors are:

- Trigger disabled. Disabling a trigger actually deletes the subscription of the trigger from the Broker. This is not the same as suspending a trigger temporarily. If you disable a trigger and then enable it, the trigger's client queue on the Broker will be deleted, even if there are documents in it. Any documents published during the period when the trigger was disabled will be discarded by the Broker, because there is no subscriber for these documents.
- Logical fault. Suppose you are synchronizing data between two databases and you have specified an insert notification on one table. You publish a document when a row is inserted into the table. A trigger subscribes that and calls a service that inserts a row into the target table. If you compare the source and target table and find some rows missing in the target table, you may presume some documents are lost. In reality, you may find that the documents were published and subscribed correctly and the service inserting data in the target table might have thrown an error (for example, it might have violated a constraint in the target side).

To troubleshoot this situation, add a checkpoint after every step. For example, you can use `pub.flow:debugLog` at various points in the service invoked by the trigger to log messages to the server log, using this information to make sure the trigger is invoking the service correctly.

PublishAndWait does not work as expected

- Matching the tag for request and reply documents: The tag envelope field is given a unique value for each `PublishAndWait` service execution. Integration Server uses this field to match reply documents to their corresponding requests. If you are replying back to a request document, make sure the tag in the reply document matches that in the request document.
- Multiple replies for a single request: Integration Server may receive multiple reply documents in response to a request but will only use the first reply document. All other replies are discarded.

Retries do not work as expected

- Trigger services using WmART-compliant adapter services are retried when the adapter run time (ART) throws **runtime** exceptions (the ART throws a runtime exception to indicate that a transient error has occurred, such as when a back-end resource is unavailable). If you handle ("catch") runtime exceptions using sequence steps in the flow code and the sequence step throws a service exception, the ART container will not retry the service. To ensure that ART retries trigger services automatically, either avoid catching runtime exceptions or throw the exceptions again as **runtime**, not **service**, exceptions. For more information about this topic, navigate to the Knowledge Base on Empower and search for the following article:
- 1611547540 (navigate to "Knowledge Center > Search The Knowledge Center" and enter this document ID in the "Search for:" "Any Document with ID" field)

Connection problems between Integration Server and Broker

Many factors can cause Integration Server to disconnect from the Broker. Common causes include:

- Broker ping property is not set. Ensure that you have set the property `watt.server.dispatcher.comms.brokerPing` to 30000 (or a reasonable value) in `server.cnf`. If this property is not set, the connection between Integration Server and Broker may be closed for being idle, in which case it will not automatically reconnect.

General Recommendations

Fixes

Review the available Integration Server Publish/Subscribe fixes. These fixes are part of the Integration Server Core fixes.

Delivery of Duplicate Documents

In certain cases, a document may be delivered twice. For example, Integration Server retrieves a guaranteed document from the Broker and processes it, but if Integration Server loses its connection to the Broker before it acknowledges the document to the Broker, the Broker redelivers the same document when the connection is re-established. To handle such cases, you can employ duplicate detection techniques. For more information about these techniques, see the *webMethods Publish-Subscribe Developer's Guide*.

Pinpoint the Location of a Problem

A lot of perceived Broker-related problems are actually occurring within Integration Server. In order to troubleshoot publish/subscribe problems effectively, try to narrow down the possible location of the problem as much as possible. For example, by adding debug statements in trigger services; increasing the logging level for facilities such as Dispatcher (0098) and Broker Transport Layer (0099); reviewing Service Usage; and monitoring Broker diagnostics like "last published/last retrieved".

Information to Gather when Opening a Support Incident

When you submit a support incident related to publish/subscribe issues, please provide the following information:

- Information listed on the [Integration Server General Information](#) page.
- Logs for logging facilities 97, 98, and 99 at TRACE level
- If applicable (for example, when a trigger service does not respond), obtain a series of Integration Server thread dumps.
Please make sure to capture thread dumps 30 seconds to 1 minute apart. Capturing multiple thread dumps is important because they are helpful in comparing the state of the threads over time.
This information provides important insight into what different threads inside Integration Server's JVM were doing.
- Integration Server Service Usage page
- Output of "netstat -an" command from the Integration Server and Broker machines
- Information from Broker Administrator:
 - Connection status of trigger clients
 - Number of sessions from the relevant triggers
 - Time when the sessions were created and the IP address and port number from which these trigger client sessions were created
- Broker version and fix level
- Broker operating system information
- Broker client API version and fix level
- Architecture setup:
 - Broker and Integration Server hosted on the same machine or different machines
 - Whether Integration Server is clustered or standalone
 - Whether triggers are clustered or not
 - Any other relevant information

JMS

In addition to the native Broker messaging, Integration Server can publish and subscribe to messages based on the JMS standard.

Common Problems

Can't enable the JMS connection

- make sure your Connection Factory is properly bound to the JNDI.
- Make sure the required client (jar) files were loaded by the Integration Server (you can check this on the Integration Server 'About' page).

Information to Gather When Opening a Support Incident

When you submit a support incident related to issues with JMS messaging, provide the following information:

- To which JMS provider your Integration server is connected
For example: wM Broker / Weblogic / Websphere / etc.
- How the JMS objects were bound into the JNDI
For example: via the MWS UI; via a script; bound via interface of 3rd Party, for example the Weblogic interface.
In the JNDI connection configuration, include the lookup of the bound objects (obtained by pressing the "play" icon).
- Which JNDI you used to bind the JMS objects
For example: the built-in webMethods JNDI; no JNDI (you are then using the webMethods Native feature); Sun File System.
- If your Integration Server is connected to a JMS provider other than the webMethods Broker, which client files (usually jar) are used.
- Which JMS model is involved; Point-To-Point (using JMS Queues) or Publish-Subscribe (using JMS Topics).
- Clear description of the problem
- The error message you get, if any
Apart from the error message, specify the side-effect, the problem that is caused. Only specifying that you are seeing an error message is usually not enough.
- The JMS connection details where the problem occurs (are all connections affected, or just one/some specific connections?)
- The stage at which the problem occurs
For example: during object binding; when enabling the connection; when running a transaction
- Integration Server Server log
- Integration Server Error log with a full stack-trace (check the "expand stack-trace" box on the error log page)
- Logs from the JMS Provider if they are available.

Integration Server - Reverse HTTP Gateway

Introduction

You can configure Integration Server to use a Reverse HTTP Gateway Server to intercept requests from external clients before passing the requests to your internal server. This allows you to isolate Integration Server behind an internal firewall.

In a Reverse HTTP Gateway configuration, you configure one Integration Server in the DMZ to be a Reverse HTTP Gateway Server and configure your Internal Integration Server to connect to the Reverse HTTP Gateway Server.

For an Integration Server to function as a Reverse HTTP Gateway Server, it must have a gateway external port to listen for requests from external clients (partners) and a gateway registration port through which it maintains its connection to the Internal Server. For security purposes, the Internal Server initiates the connections to the Reverse HTTP Gateway Server's registration port.

Diagnostics

Once your Reverse HTTP Gateway has been configured (one or more Integration Server acting as the "Gateway Server," and one or more Integration Server acting as the "Internal Server"), currently the best means for verifying the health of the registered, reverse connections is by making use of the `netstat` command. Assuming a Gateway Server with registration port 6000, here is what a `netstat` command would reveal from each host:

Gateway Server

```
D:\>netstat -an | grep 6000
TCP 0.0.0.0:6000 0.0.0.0:0 LISTENING
...
TCP dmz.network.com:6000 internal.network.com:3639 ESTABLISHED
TCP dmz.network.com:6000 internal.network.com:3723 ESTABLISHED
TCP dmz.network.com:6000 internal.network.com:3724 ESTABLISHED
TCP dmz.network.com:6000 internal.network.com:3725 ESTABLISHED
TCP dmz.network.com:6000 internal.network.com:3726 ESTABLISHED
```

Internal Server

```
D:\>netstat -an | grep 6000
TCP internal.network.com:3639 dmz.network.com:6000 ESTABLISHED
TCP internal.network.com:3723 dmz.network.com:6000 ESTABLISHED
TCP internal.network.com:3724 dmz.network.com:6000 ESTABLISHED
TCP internal.network.com:3725 dmz.network.com:6000 ESTABLISHED
TCP internal.network.com:3726 dmz.network.com:6000 ESTABLISHED
```

The number of ESTABLISHED connections should not exceed the number specified on the Max Connections parameter on the Internal Server's Registration Internal port page (available from Integration Server Administrator > Security > Ports).

Common Problems

Establishing and Maintaining Reverse Connections

A Reverse HTTP Gateway configuration by nature spans multiple network segments and typically crosses firewalls. These firewalls can have a variety of different policies and mechanisms for severing persistent connections across it. When the reverse connections are initially enabled from the Internal Server, if the connection fails, the following message will be seen on the Internal Server's server log:

```
2010-04-02 14:38:56.218 EDT [ISP.0046.0019C] Unable to establish connection to the Reverse Gateway
server dmz.network.com:6000, Exception --> Connection refused: connect.
2010-04-02 14:39:47.134 EDT [ISP.0046.0019C] Unable to establish connection to the Reverse Gateway
server dmz.network.com:6000, Exception --> Connection refused: connect.
2010-04-02 14:40:38.144 EDT [ISP.0046.0019C] Unable to establish connection to the Reverse Gateway
server dmz.network.com:6000, Exception --> Connection refused: connect.
```

Notice the interval between messages is approximately 51 seconds. This interval will vary from environment to environment, and is equal to 30 seconds plus some variable overhead (depending on Integration Server load, thread scheduling, time waiting for a connection from the network, DNS, etc). These same messages will be seen if an already established connection is lost (for example, lost due to a restarted Gateway Server).

Also, once the connections are established, the Internal Server will send a keepAlive packet every 60 seconds. The intention is to prevent a firewall from identifying this as an idle connection that should be terminated.

At this time, none of these intervals are exposed or configurable to the end-user. This should be improved in a future version.

Internal Server Port Number as Unique Key

Currently, from an Internal Server, when establishing a connection (or set of connections) to a given Gateway Server, you do this via the Integration Server Administrator > Security > Ports page. After you've enabled such connections, the Integration Server will display this as a listening port (even though that port number is not truly in use on the internal host).

For example, for a Gateway Server on host A with proxy (4000) and registration (6000), when you look at its IS Admin > Ports table, you'd see 4000 (gateway external), 6000 (gateway registration), and 5555 all enabled, and a netstat would show them all in LISTENING state. But for an Internal Server on host B that is connected to the above Gateway Server on host A, the IS Admin > Ports table would show 6000 (registration internal) and 5555, and a netstat would only show 5555 in LISTENING state. On host B, these connections are actually initiated from some range of ports (such as 3639-3726 in the previous section) for outgoing connections.

One resulting inconvenience of this is that when connecting a given Internal Server to multiple Gateway Server instances, the registration ports on those Gateway Servers must use different port numbers. It is the same effect as trying to define two listening ports on the Internal Server with the same port number.

This will be improved in a future version in the near future, but as of 8.0.x, this is the behavior.

Load Balancing

The load balancing algorithm used in Integration Server 7.1.x and later has changed from prior versions (Reverse Invoke). In Reverse Invoke, the reverse gateway used a purely round-robin mechanism among all reverse connections. But with Reverse HTTP Gateway, the Gateway Server now uses a least-busy mechanism (specifically it looks at each of the Internal Server instances connected to its registration port, determines which has the most free sockets, and then forwards the request to a socket of that instance).

Integration Server - Scheduler

Common Symptoms

Any of the following symptoms may indicate that there could be an issue related to the Integration Server Scheduler:

- Cluster Target Node option shows "Cluster unavailable"
- Next run shows 0 and status shows Running
- Next run shows 0 and status shows Active
- Tasks not executing at expected times
- Cluster aware tasks running on multiple nodes at the same time
- No threads available to run Cron task
- Server.log shows Scheduler: Resources unavailable: Rolling back due to scheduler thread throttle reached:nn (where nn is the value set for Scheduler Thread Throttle)

Possible Causes and Resolutions

Cluster Target Node Option Shows "Cluster unavailable"

The entire scheduled task (metadata and execution timing) is stored in the database identified by the ISInternal JDBC connection pool. The Task data can be found in the IS_USER_TASKS table. Scheduled tasks can be used in clustered and non-clustered environments. When dealing with scheduled tasks in a clustered environment, please make sure that the ISInternal JDBC connection pools of all the cluster nodes are pointing to the same database schema.

Next Run Shows 0 and Status Shows "Running"

This symptom occurs if a task is marked for Repeat From End of Invocation and the task is still running. Once the running task completes, next Run resets back to a normal value. In this situation, you may need to troubleshoot the task service for possible delays and bottlenecks.

Next Run Shows 0 and Status Shows Active

This situation can occur if no more threads are available in the Scheduler Thread Pool. When this situation occurs, the Integration Server server log will show "Scheduler: Resources unavailable: Rolling back due to scheduler thread throttle reached:30". This message may be a configuration issue, which can be corrected by setting the appropriate value for the Scheduler Thread Throttle parameter under "Settings > Resources > Edit Resource Settings".

Another possibility is that all scheduler threads are stuck, due to which the scheduler thread will be in use and will not be returned to the pool. (Thread dumps will be helpful to confirm this situation.)

Tasks Not Executing at Expected Times

This symptom may be related to the `"watt.server.scheduler.maxWait"` parameter. In a clustered environment, please make sure that all nodes of the cluster have the same value set for this parameter. For more details, see *"webMethods Integration Server Administrators Guide"* or *"Administering webMethods Integration Server"*.

Cluster Aware Scheduler Task Running on Multiple Nodes at Same Time

This symptom can occur if the clocks of the systems hosting the Integration Server nodes are not in sync. Please make sure that clocks of all the nodes, that are part of the cluster, are in sync.

No Threads Available to Run Cron Task

Mainly there are four Cron Daemons that exist in Integration Server for scheduling tasks (each daemon name is shown in parenthesis below):

4. Cron Daemon for scheduling system tasks (Cron Daemon)
5. Cron Daemon for scheduling user tasks in a non-clustered environment (User Tasks)
6. Cron Daemon for scheduling user tasks in a clustered environment (User Cluster Tasks)
7. Cron daemon for scheduling adapter notifications (Cron Daemon)

Each thread daemon has a thread pool associated with it. In order for Cron Daemon to run the scheduled task, it should have a thread available in its thread pool.

- The maximum threads in the Cron Daemon pool is configured on the `"watt.server.cronMaxThreads"` parameter (used in scenario 1 and 4 above). Default value is 5.
- The remaining two (scenarios 2 and 3) are configured on the `"watt.server.scheduler.threadThrottle"` parameter (or from Integration Server Administrator > Settings > Resources > Scheduler Thread Throttle).
The watt parameters are set in the server.cnf file. For details about these parameters, see *"webMethods Integration Server Administrators Guide"* or *"Administering webMethods Integration Server"*.

General Recommendations

When troubleshooting Scheduler problems, you should also follow the Integration Server troubleshooting recommendations, which can be found in [General Recommendations](#).

Information to Gather When Opening a Support Incident

To troubleshoot Integration Server cluster related issues, obtain the following information:

- Provide the information listed on the [Integration Server General Information](#) page
- Screen shot of the Integration Server scheduler showing the scheduled jobs
- Whether the task was single, repeating, or complex

Integration Server - Session

Diagnostics

Session Log

Session activity (such as Session Started and Session Ended) is written to the Session log. By default, this log is stored in the <IntegrationServer-directory>/logs/session.log file. By virtue of enabling the ISCoreAudit JDBC Pool, this log will be kept in the WmSession table of the ISCoreAudit database schema. The log contents can also be viewed via the Integration Server Administrator > Logs > Session page. Below is a sample session log:

```
2010-04-02 10:24:31.898 EDT mcmuser:5555 Administrator 127.0.0.1 Ended 15 83080
42d27f203e6311df95b7d251439f4008
2010-04-02 10:24:31.882 EDT mcmuser:5555 Administrator 127.0.0.1 Expired 15 83064
42d27f203e6311df95b7d251439f4008
2010-04-02 10:23:08.818 EDT mcmuser:5555 Default 127.0.0.1 Started 0 0
42d27f203e6311df95b7d251439f4008
```

Server Log

Session creation and user authentication can also be seen in the server log by increasing the IS Admin > Settings > Logging > Integration Server > 0012 Authentication facility to TRACE. This will display information like the following during incoming requests:

```
2010-04-02 10:23:08.896 EDT [ISS.0012.0006T] Client provided session id 42XXXXXX008 and that
session exists
2010-04-02 10:23:08.896 EDT [ISS.0012.0008T] Skipping authentication because session 42XXXXXX008
contains user (Administrator)
```

Server.cnf File

The <IntegrationServer-directory>/config/server.cnf file contains the following session-related control parameters. Some are only relevant when running in a clustered Integration Server environment.

```
watt.server.clientTimeout=10
...
watt.server.cluster.sessTimeout=60
watt.server.cluster.purgeSessions=5
watt.server.cluster.updateSession=false
```

Common Problems / Typical Causes

Background / Internals

Sessions are stored in Integration Server memory. They are lightweight by default, but depending on the type of request, a high session count may be an indicator of an imminent resource problem, such as a high number of JVM threads or high memory usage.

Sessions are created when a request hits the Integration Server. If the connecting client explicitly disconnects from the server, the session will be deleted immediately. Otherwise, the service's stateless property will come into play. For a stateless service, the session will be deleted as soon as service execution completes (that is, perhaps in a matter of seconds). But for a stateful service invocation, the session will remain on the server after service execution completes until it reaches the value specified on the `"watt.server.clientTimeout"` parameter. After this point, the 'Session Sweeper' will clean it up.

Sessions are governed by the `"watt.server.clientTimeout"` parameter, for which the default is 10 minutes. This timeout value can be set from the Integration Server Administrator from either `Settings > Resources > Session Timeout`, or from `Settings > Extended Settings` (like all other `server.cnf` parameters). When a new session is created, on the `IS Admin > Server > Statistics > Total Sessions` page, you will see its 'Session Expires' value count down from 600 (10 minutes) to 0. Once it reaches 0, it will display the text "expired". All sessions listed as "expired" are eligible to be swept by the 'Session Sweeper'. The sweeper is a System Scheduler Task that can be seen on the `IS Admin > Server > Scheduler > View System Tasks` page. This task will be an interval of the same length as what was set in `"watt.server.clientTimeout"` (which is 10 in the same example). Once this task runs, all expired sessions will be removed. So in reality, a session could sit in memory for anywhere from 1 to 2 times the `"watt.server.clientTimeout"` value, depending on timing. If a subsequent request had been made on a given session while it was counting down from 600 to 0, the 'Session Expires' value would reset to 600.

When Integration Server clustering is used, every time a request is received by one of the server nodes, the local session is still created in memory, but a shared session is also created in the cluster cache (Oracle Coherence). The life of this shared session is governed by the `"watt.server.cluster.sessTimeout"` parameter (default 60 minutes), and the 'Remove Expired Cluster Sessions' System Scheduler Task will run on each server at the interval specified on the `"watt.server.cluster.purgeSessions"` parameter (default 5 minutes). The `"watt.server.cluster.updateSession"` parameter (default "false") indicates whether the server nodes will update the shared session object each time the local session is modified. This updating is typically not necessary, and is a very expensive operation performance-wise, thus it is disabled by default.

High Session Count

Every Integration Server license key has a 'max concurrent session limit' built into it. For "unlimited usage" license keys, the ceiling is actually 10,000 sessions. A session count approaching this limit is usually an indicator of some other problem. In fact, any time a session count peaks over 500, it is probably a good idea to verify that these are expected bursts in server traffic / requests.

One common cause of an unexpected, high session count is the use of a Load Balancer or monitoring tool that is sending 'ping' or 'isAlive' health check requests to the Integration Server. The best-practice method for this is to have these tools target a stateless IS service (such as a user-created, wrapper to `wm.server:ping`).

Information to Gather When Opening a Support Incident

Before you open a Support Incident, please do the following:

- Provide a screenshot of the list of current sessions. If the Integration Server Administrator is slow or not responsive, you can run the following service independently in a browser to collect the same data (`http://host:5555/invoke/wm.server.query:getSessionList`).
- Provide the `server.cnf` file OR run `http://host:5555/invoke/wm.server.admin/getDiagnosticData`, which will package a number of logs, environment, and configuration data, including the `server.cnf`.

Broker - General Troubleshooting

Troubleshooting Information

The following information can help when troubleshooting Broker problems.

Determine the Broker Server Version and Fix Level

You can use either of the following ways to determine the version of the Broker Server and its fix level that you're currently running.

- To use a command line to see the Broker Server version and fix level, run the following command:

```
<webMethods_install_directory>/Broker/bin/awbroker -version
```

- To use My webMethods Server to see the Broker Server version and fix level:
Log in to My webMethods Server, and navigate to Administration > Messaging > Server. From this page, you should see your Broker Server. If you do not see your Broker Server, click the "Add" button. The version and fix level of this Broker Server will be listed in the Version column of this page.

The Broker Server version number is displayed in this format:

```
7.1.2.0.114_Fix3 041209
```

- Where:
 - 7.1.2 represents the main version number (in this case: Broker 7.1.2)
 - x.x.x.0 shows the Service Pack number applied
 - 114 represents the build number
 - Fix3 is the current Broker Core Fix being applied
 - 041209 represents the build date (in this case, April 12, 2009)

Determine the Broker Java API Version

Any webMethods product that is communicating or connecting to the Broker, such as Integration Server or My webMethods Server, or any custom application, is using the Broker API library to communicate with the Broker. There are some cases where the cause of an issue may not be in the Broker itself, but may be in the Broker API. In such cases, it's good to find out what Broker API fix level you're currently running with to help troubleshooting the issue.

To determine what Broker Java API version and fix level is being used, perform the following steps:

- Go to <webMethods_install_directory>/common/lib directory and open wm-brokerclient.jar
- Locate the file COM/activesw/api/client/version.properties and open version.properties file
- The API version info is shown in the following format:

```
Version.Date=091208
Version.Major=7
Version.Minor=1
Version.Number=7.1.2.0.78
Version.Label=
```

Where:

- Version.Number indicates the version is 7.1.2
- Version.Date indicates the date that this API fix was released. From the example above, we know this is the Broker API 7.1.2 GA release without any fix

In general, given a Broker API library (wm-brokerclient.jar), to truly know which fix has been applied, you would need to compare its version.properties > Version.Date against the various released fixes' version.properties > Version.Date values (such as BR_7.1.2_Java_API_Fix1 or BR_7.1.2_Java_API_Fix2). Alternately, you could send a request to Software AG Global Support to confirm the version and fix level.

Within a java program, to obtain the Broker API version, use the method
BrokerClient.getApiVersionNumber().

Look For Errors or Messages in the Broker Server Log

The Broker Server log records operational and error messages issued by Broker Monitor and Broker Server. You can refer to this log to find out when a Broker Server started running or stopped running. You can also refer to the log to examine error messages, warnings, and information messages issued by Broker Server while it was running.

The log consists of two files. Both files reside on the machine where Broker Server is installed.

- Broker Server writes its log entries to <Broker_Server_Data_Directory>/logmsgs
- Broker Monitor writes its log entries to <Broker_Server_Data_Directory>/logmsgs.mon

You can view the entries in both of these logs in My webMethods Server: Administration > Messaging > Servers > <your_Server_Name> > Server Log

Besides logging messages in the Broker Server log, Broker Monitor and Broker Server can optionally send their messages to the host machine's native logging facility (Event Log Service on Windows or syslog on Unix).

You can view these logs entries by doing the following:

Windows

- Start the Windows Event Viewer: Start > Settings > Control Panel > Administrative Tools > Event Viewer.
- webMethods messages are stored in the Application log.

UNIX/Linux

webMethods messages are stored in the following files:

Solaris

- /var/log/broker.alert
- /var/log/broker.info

AIX

- /var/opt/webMethods/broker.alert
- /var/opt/webMethods/broker.info

HP-UX

- /var/adm/syslog/broker.alert
- /var/adm/syslog/broker.info

Note: If the log files are not in their default location, view the `/etc/syslog.conf` file to determine their location.

Logs can be very large. Typically, you want to filter the log to extract a certain group of messages, such as all messages for a particular day. For example, to extract errors that occurred on June 4th, you would type the following at a command-line prompt:

```
grep "Jun 4" broker*
```

This command displays all lines containing the string **Jun 4** in all files whose names start with **broker** (for example, `broker.alert` and `broker.info`). Be sure to type the date string in the format you use to represent the date.

Start the Broker Server in Debug Mode

For troubleshooting, you may want to start a Broker Server in debug mode to gather more debug information. To do so, start the Broker Server from the command-line prompt using the following command:

```
<webMethods_install_directory>/Broker/bin/awbroker -d <full_path_of_BrokerServer_data_directory> -
blah
```

For example:

```
/opt/webMethods/Broker/bin/awbroker -d /var/opt/webmethods/awbrokers71/default -blah
```

Broker Queue Storage

Understanding Broker Queue Storage

webMethods Broker stores Broker Server configuration data, Broker metadata - such as Broker names, client groups and document types - and run-time data in the file system. By default, the Broker data files are located in the Broker Server data directory.

- <webMethods_install_dir>\data\awbrokers82\default on Windows
- /var/opt/webMethods/awbrokers82/default on Unix

There are two different types of Broker queue storage, Combined storage and Split storage. The user can decide which type to use when creating a Broker Server. By default, a Broker Server is created using Split storage.

For more detailed information, please refer to the "*Administering webMethods Broker*" Guide.

Contents of the Broker Server data directory

The following files are generated when the Broker Server is created.

Configuration files

- awbroker.cfg
This is the main configuration file of the Broker Server. It contains a number of configuration settings, including:
 - the location of BrokerConfig.js and BrokerData.js,
 - the location of license.xml,
 - the location of the awbroker executable,
 - logging settings.
 If this file is missing or corrupt, the Broker Server will fail to start.
- basicauth.cfg
This file only exists since Broker version 8.2. It contains the configuration for basic authentication with LDAP and ADSI.

- **license.xml**
This file contains the Broker license key. Without this file, the Broker Server will fail to start.

Metadata storage files

- **BrokerConfig.qs**
This is the configuration file of the metadata storage. It contains the locations, reserved size and maximum size of the BrokerConfig.qs.stor and BrokerConfig.qs.log files. This file has a binary format which can only be viewed using the "strings" or "od -c" commands on Unix.

```
strings BrokerConfig.qs
od -c BrokerConfig.qs
```

Some text editors can display the contents in hexadecimal mode. Be careful not to edit. If this file is missing or corrupt, the broker server will fail to start.

- **BrokerConfig.qs.stor**
All the metadata from the Broker Server is stored in this file. The following objects are considered metadata: Brokers, territories, gateways, document types, client groups, clients, SSL configuration and more.
The size of this storage file can be up to 32GB. The size of an existing *.qs.stor file can be increased and more than one *.qs.stor file can be created; up to 64 files.
If this file is missing or corrupt, the broker will fail to start.
- **BrokerConfig.qs.log**
This file is used as transitional storage - the first phase of the two-phased commit storage mechanism. All metadata creations, updates or changes are written to this file before the Broker asynchronously writes them to BrokerConfig.qs.stor - the second phase of the storage mechanism.
The size of BrokerConfig.qs.log can't be increased once it's created. If you need a larger *.qs.log file, you create a new *.qs.log file with a different name and a bigger size. The Broker will use with the latest *.qs.log file that has been created.

Note: If all the BrokerConfig.qs.* files are removed or missing, the Broker Server is able to start, because it will automatically create new BrokerConfig.qs.* files with the default file names and sizes. All metadata/configuration information from this Broker Server - such as Broker names, client groups and document types - are gone then.

Run-time data storage files

These files do not exist when a Broker Server is created with the combined storage type.

- **BrokerData.qs**
This is the configuration file of the run-time data storage. It contains the locations, reserved size and maximum size of the BrokerData.qs.stor and BrokerData.qs.log files.
- **BrokerData.qs.stor**
All the run-time transactions/documents are stored in this file until they are acknowledged.
The size of this storage file can be up to 32GB. The size of an existing *.qs.stor file can be increased and more than one *.qs.stor file can be created; up to 64 files.
If this file is missing or corrupt, the broker will fail to start.

- BrokerData.qs.log

This file is used as transitional storage - the first phase of the two-phased commit storage mechanism. All documents are written to this file before the Broker asynchronously writes them to BrokerData.qs.stor - the second phase of the storage mechanism.

The size of BrokerData.qs.log can't be increased once it's created. If you need a larger *.qs.log file, you create a new *.qs.log file with a different name and a bigger size. The Broker will use with the latest *.qs.log file that has been created.

The size of the BrokerData.qs.log file determines the maximum size of a transaction (or batch of transactions) can be accepted by the Broker.

Note: If all the BrokerData.qs.* files are removed or missing, the Broker Server is able to start, because it will automatically create new BrokerData.qs.* files with the default file names and sizes. All run-time transactions/documents that were queued for the Broker's clients that is belonging to this Broker Server are removed/gone as well.

Debug files

- awbroker.pid

This text file contains the process id of awbroker that is currently running.

- diag.log

This file contains diagnostic information, which is written when the Broker server runs into a problem. This file can be analyzed by SoftwareAG to determine if there is an issue with the Broker.

- Logsmgs

This file contains log messages from the Broker Server. Three severity levels are logged: "information", "warning" and "error". The contents of this file can be viewed in My webMethods Server > Administration > Messaging > Broker Servers > Servers > Broker Server Details > Server Log.

- cbtrace.store

This file contains diagnostic traces, which are written when the Broker server runs into a problem. This file can be analyzed by SoftwareAG to determine if there is an issue with the Broker.

Possible Causes of Broker Server Out-of-Storage Errors

The Broker Server will throw the following exception when it doesn't have enough space within its storage file(s).

```
"(100-2015): The operation failed because the Broker does not have enough storage to process the request. Alert your administrator"
```

This error indicates that there is not enough free space in either BrokerData.qs.stor or BrokerData.qs.log; or in BrokerConfig.qs.stor or BrokerConfig.qs.log when using combined storage or when the error happens during a configuration update. Due to this, the Broker is unable to handle the incoming request. You can confirm by checking your Broker Server's Utilization to see how much space of Configuration and Run-time Data storage has been used and how much is still available.

One of the following four scenarios can be the main cause of Broker Server Out-of-Storage error.

Scenario 1: Published document (or batch of documents) exceeds the maximum transaction size

Guaranteed documents are written to the Broker storage in two phases: first to BrokerData.qs.log and then to BrokerData.qs.stor. The size of BrokerData.qs.log determines how much data can be stored by the Broker in a single transaction. One or more documents can be published to the Broker within a single transaction context; this is determined by the client. If the total size of the document(s) that the client publishes within a single transaction exceeds the size of the BrokerData.qs.log file, the Broker cannot accept the transaction and will throw an Out-of-Storage exception.

How to verify and what to check

- Refer to [Calculating the size of a Broker document "on the wire"](#) to calculate the size of the documents;
- Check the implementation of the Broker client to check how it handles transactions when publishing.

Suggested solution

- Reduce the transaction size by publishing fewer documents within the same context; or
- Increase the maximum transaction size of the Broker by generating a new BrokerData.qs.log file. Use the command line tool server_config to generate a new BrokerData.qs.log file with a new name.

Scenario 2: Documents are stuck in client queue(s)

Guaranteed documents will occupy space in the Broker Server storage until the client that subscribed to them acknowledges them. Once the document is acknowledged, the broker will remove/delete the document from the queue and free up the storage space.

If the Broker client - for example an Integration Server trigger - does not process or acknowledge the documents for some reason, the documents are stuck in the client queue. If this situation lasts for a longer time, the Broker Server will eventually run out of space. How quickly the Broker runs out of space depends on

- the size of the storage file(s);
- the size of the documents (see also [Calculating the size of a Broker document "on the wire"](#));
- the number of documents queued;
- the amount of time since the client last acknowledged the documents it processed.

Note: if multiple clients are not acknowledging documents of the same document type, the Broker does not duplicate the documents for each client.

Cause #1: Client Is Not Connected

When the Broker client is not connected to the Broker, the documents can't be processed by the client and remain in the client's queue. If the publisher keeps publishing documents to which this client subscribes, the documents will pile up in the client queue.

For example, Integration Server #1 has a trigger which subscribes to document type 'A', and this Integration Server is down for several hours. Integration Server #2 is still publishing documents with type 'A' while Integration Server #1 is down. This will cause all the published documents of type 'A' to pile up in the trigger's queue, waiting until the trigger in Integration Server #1 reconnects, processes the documents and acknowledges them.

How to verify and what to check

- Check all the Broker clients' queues in My webMethods Server
 - Navigate to Administration > Messaging > Broker Servers > Clients and look for clients that have documents in their queue. Hint: sort the search result by "Docs Queued" descending.
 - Check if the clients that have documents in their queues are currently connected to the broker. Clients that are not connected can be the cause of the out-of-storage issue.

Suggested solution

- Determine why the client is not connected to the broker. There could be a problem with client itself, or there could be a network issue. A client that is connected to the broker will normally retrieve the documents from its queue and therefore not fill up the Broker Server storage.
- If it's expected that the client is not connected, and the publisher can't stop or pause publishing documents, you need to increase the Broker Server storage in order for the Broker to have sufficient space to process further requests. Use the command line tool `server_config` to increase the `BrokerData.qs.stor` or add an additional `*.qs.stor` file.

Cause #2: Client Doesn't Drain the Queue

When the Broker client is successfully connected to the Broker, but not draining the documents from its queue, the queue keeps growing and no documents are processed.

Integration Server trigger in "Suspended" state

How to verify and what to check

- If the Broker client is an Integration Server trigger, check if the trigger is currently in "Suspended" state.
"Suspended" means that the trigger is enabled and has an active subscription for the document type, but it is currently not retrieving and/or processing documents from its queue.
 - In Integration Server Administrator, navigate to Settings > Messaging > Broker/Local Trigger Management.
On this page, check the "Active" columns under "Document Retrieval" and "Document Processing". If you see "No" displayed, the trigger has its Document Retrieval and/or Processing suspended. Verify in My webMethods Server Administration > Messaging > Broker Servers > Clients if that trigger matches the Broker client that has documents in its queue. Suspended trigger clients can be the cause of the out-of-storage issue.

Suggested solution

- Determine why the trigger is in "Suspended" state.
- Activate/resume the trigger to allow it to retrieve and process documents.
- If it's expected that the trigger is suspended, and you can't activate/resume it at this point in time, you need to increase the Broker Server storage in order for the Broker to have sufficient space to process further requests.
Use the command line tool `server_config` to increase the `BrokerData.qs.stor` or add an additional `*.qs.stor` file.

Integration Server trigger has a locked queue

How to verify and what to check

- If the Broker client is an Integration Server trigger, check if the client queue is locked.
If the queue is locked, the client will not be able to retrieve any documents.
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Clients and locate the trigger client that is not draining documents from its queue. Click on the "Browse Queue" tab and check what is displayed for "Queue is Locked".

Suggested solution

- If the client queue is in the locked state, you need to unlock it in order for the documents to be retrieved and processed.
- If it is expected that the queue is locked, and you can't unlock at this point in time, you need to increase the Broker Server storage in order for the Broker to have sufficient space to process further requests.
Use the command line tool `server_config` to increase the `BrokerData.qs.stor` or add an additional `*.qs.stor` file.

Integration Server trigger doesn't invoke flow service

How to verify and what to check

- If the Broker client is an Integration Server trigger, check if the configured flow service is correctly invoked.
If the trigger does not invoke the flow service, or the flow service gets stuck, the documents that the trigger is processing will not be acknowledged to the Broker.
 - In the Integration Server Administrator, navigate to Server > Service Usage.
On this page, check the flow service which is invoked by the trigger.
 - A number value between parentheses behind the service name indicates that the flow service is currently running. The number stands for the thread count.
 - The value in the "Last Ran" column shows the most recent time when service completed.
 - With this information you can determine whether the flow service was ever invoked by the trigger. If you have multiple Integration Servers in a cluster, you need to check this on each of the Integration Server cluster nodes.
 - If the service is shown as currently running and the "Last Ran" column is not updated, then the flow service is likely unable to complete fast enough. The trigger's document processing will be delayed, which in turn causes the documents to pile up in the trigger's queue. This is commonly caused by slow a backend application, such as a database, ERP or CRM system. In order to find the root cause, investigate the backend application or debug the flow service until you find the bottle neck.

Suggested solution

- Restart the Integration Server to terminate any hanging threads.
Note: in order to investigate the cause of the hang, refer to [Integration Server - Hang](#).
- Once the documents are processed, the Broker Server storage will be freed up. If this is taking too long, you need to increase the Broker Server storage in order for the Broker to have sufficient space to process further requests.
Use the command line tool `server_config` to increase the `BrokerData.qs.stor` or add an additional `*.qs.stor` file.

Serial trigger client is waiting for acknowledgement of the first document

How to verify and what to check

- If the Integration Server trigger has Processing Mode "Serial", check if the first document in the queue is blocking the rest.
 - On the Integration Server Service Usage page, the flow service invoked by the trigger is not shown as currently running, and the "Last Ran" time is not up to date.
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Clients and locate the trigger client that is not draining documents from its queue. Click on the "Browse Queue" tab and click on the "Start" button to retrieve the first few documents from the queue. If the first document in the queue has the status "Unacknowledged", this document was not acknowledged for some reason. It remains in the queue and because the trigger is has Processing Mode "Serial" all documents after it are blocked.

Suggested solution

- Disconnect the session that has the oldest "Last Activity" time from this trigger client
On the Client Details Page in My webMethods Server, click on the "Sessions" tab. Select the checkbox in front of the session and click the "Disconnect" button. A new connection will be created and the first document in the queue will be reprocessed.
Note: If you have more than one Integration Server in a cluster connecting to the Broker using the same client-prefix, you should see more than one session on the "Sessions" tab of the client. The number of sessions should match the number of the Integration Server instances in the cluster.
- Restart the Integration Server.
If disconnecting the session(s) doesn't help, or if you are not sure which one to disconnect, you can choose to restart the Integration Server.
- Once the documents are processed, the Broker Server storage will be freed up. If this is taking too long, you need to increase the Broker Server storage in order for the Broker to have sufficient space to process further requests.
Use the command line tool `server_config` to increase the `BrokerData.qs.stor` or add an additional `*.qs.stor` file.

Stale Broker Client Session

How to verify and what to check

- If a Broker client is not retrieving documents from the queue, but the client itself appears to be fine, the documents in the queue could be associated with a stale session.
If the connection between the client and the Broker was disrupted, it may happen that the client reconnects with a new session, but the Broker did not detect the disconnection of the old session. Documents that are associated with the old session, will remain in the queue, because the old session is not used by the client.
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Clients and locate the client that is not draining documents from its queue. Click on the "Sessions" tab of this client and check the "Last Activity" time. Or click on the "Statistics" tab and check the "Last Retrieved" time.
If the "Last Activity" or "Last Retrieved" time is not up to date and there are currently documents in the queue, the session can be considered stale.

Suggested solution

- Disconnect the session that has the oldest "Last Activity" time from this trigger client
On the Client Details Page in My webMethods Server, click on the "Sessions" tab. Select the checkbox in front of the session and click the "Disconnect" button. A new connection will be created and the first document in the queue will be reprocessed.
Note: If you have more than one Integration Server in a cluster connecting to the Broker using the same client-prefix, you should see more than one session on the "Sessions" tab of the client. The number of sessions should match the number of the Integration Server instances in the cluster.
- Restart the Integration Server.
If disconnecting the session(s) doesn't help, or if you are not sure which one to disconnect, you can choose to restart the Integration Server.
- Restart the Broker
If restarting the Integration Server(s) doesn't help, you can restart the Broker. On startup, the Broker will clean up the sessions.
- Once the documents are processed, the Broker Server storage will be freed up. If this is taking too long, you need to increase the Broker Server storage in order for the Broker to have sufficient space to process further requests.
Use the command line tool `server_config` to increase the `BrokerData.qs.stor` or add an additional `*.qs.stor` file.

Cause #3: Publishers Outperform Subscribers

When the publishing clients are publishing guaranteed documents a lot faster than the subscribing clients are processing these documents, the documents will pile up in the subscribers' queues. If this situation lasts for a longer time, the Broker Server will eventually run out of space. How quickly the Broker runs out of space depends on

- the size of the storage file(s);
- the size of the documents (see also [Calculating the size of a Broker document "on the wire"](#));
- the number of documents queued;
- how much faster the documents are published than they are processed.

How to verify and what to check

- A common case for the subscriber not processing the documents fast enough is when the "Shared Document Order" set to "Publisher".
This means the documents need to be processed in the order in which they are received from each respective publisher. The effect is that documents are processed in serial mode, one by one per publisher, and can't be processed in parallel.
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Clients and locate the client that is not draining documents from its queue. Click on the "Configuration" tab of the client and check what is displayed for "Shared Document Order". The value will be "Publisher" or "None".

Suggested Solution

- Slow down publishing
If you require the documents to be processed in the same order as they are published, the "Shared Document Order" should be set to "Publisher" for the subscribing client. Then the serial mode processing is expected by your design.
 - To avoid the Broker running out of storage in this situation, you have to slow down the publishing client(s). If this is not possible, you need to increase the Broker Server storage in order for the Broker to have sufficient space to process further requests.
Use the command line tool `server_config` to increase the `BrokerData.qs.stor` or add an additional `*.qs.stor` file.
- Set "Shared Document Order" to "None"
If your design does not require the documents to be processed in the same order as they are published, you can reconfigure your client to use "Shared Document Order" "None".
Note: if your subscribing client is an Integration Server trigger, you need to reconfigure your trigger using Developer/Designer. In the trigger properties, change the processing mode to "concurrent". The trigger client will be recreated with "Shared Document Order" set to "None".
Note: be careful! Recreating the Broker client will discard any documents that are currently in the queue, and it will recreate the document type subscriptions. If the queue is not empty, or the publisher is not suspended, you may lose data!

Scenario 3: When "Document Type Logging" is enabled

When "Document Type Logging" is enabled on any Broker within the Broker Server, documents get stored in the storage for the purpose of logging them to the audit database. If the Logging Utility (WmLogUtil package) on the Integration Server is not running or not able to process documents from the Broker into the Integration Server audit database, documents will remain in the Broker Server storage and occupy space.

If this situation lasts for a longer time, the Broker Server will eventually run out of space. How quickly the Broker runs out of space depends on

- the size of the storage file(s);
- the size of the documents (see also [Calculating the size of a Broker document "on the wire"](#));
- the number of documents queued.

For more details about Document Type Logging and the Logging Utility (WmLogUtil package), navigate to the Knowledge Base on Empower and search for the following article:

- 1614322261 (navigate to "Knowledge Center > Search The Knowledge Center" and enter this document ID in the "Search for:" "Any Document with ID" field)

How to verify and what to check

- If "Document Type Logging" is enabled, the logged documents need to be processed by the Logging Utility.
Check if "Document Type Logging" is enabled on any Broker within the Broker Server that has the storage issue. If "Document Type Logging" is enabled, make sure the Logging Utility (WmLogUtil) on the Integration Server is enabled and running.
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Brokers and click on the name of each Broker in this Broker Server. On the Broker Details Page, check if the checkbox is enabled for "Document Type Logging".
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Brokers and click on the name of each Broker in this Broker Server. On the Broker Details Page, click on the "Statistics" tab and check the value of "Document Log Length".
 - In the Integration Server Administrator, navigate to Packages > Management and click on the Home icon of the WmLogUtil package. On the WmLogUtil page, check if the Status is "Running"; use "Refresh Status" if needed.
- Verify whether there are any documents in the Broker's log queue, using either of the following flow services from the WmLogUtil package.
 - *pub.loggingUtility.util:brokerLoggingLength*
This service shows how many documents are currently in the Broker's log queue. These documents have not (yet) been retrieved by the Logging Utility for processing into the Integration Server audit database.
 - *pub.loggingUtility.util:brokerLoggingPeek*
This service shows the documents that are in the Broker's log queue. Use the parameter NumberToPeek to specify how many documents are shown.
- In the Integration Server Administrator, navigate to Settings > JDBC Pools and test the Pool Alias that is associated with "ISCoreAudit" using the green "Play" icon.
If there are issues with the JDBC connection to the Integration Server audit database, these need to be resolved in order for the Logging Utility to work correctly.

Suggested solution

- (Re)start the Logging Utility
If the Logging Utility is not enabled or not running, start it from the WmLogUtil page in the Integration Server Administrator.
 - In the Integration Server Administrator, navigate to Packages > Management and click on the Home icon of the WmLogUtil package. Use the "Start" link to start the Logging Utility. Use the "Stop" link first if the Logging Utility is not stopped.
- Clean the log queue
If "Document Type Logging" was enabled by accident, and you do not intend to log the documents in the Integration Server audit database, turn off "Document Type Logging" and clean the Broker's log queue.
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Brokers and click on the name of the Broker. On the Broker Details Page, uncheck the checkbox for "Document Type Logging".
 - Call service `pub.loggingUtility.util:brokerLoggingSingleExtract` as many times as the number returned by `pub.loggingUtility.util:brokerLoggingLength`. Create a new flow service that first retrieves the number of documents in the log queue, then loops that many times and extracts a document in each iteration of the loop. You can choose to save the documents somewhere, or to discard them.
Note: be careful! It is not recommended to perform this in a production environment. You should carefully assess the performance impact of your custom service on both the Broker and Integration Server.
- Recreate the Broker Server storage
The log queue is an internal queue which is not viewable, removable, or accessible from the My webMethods Server interface. If you need to delete the documents from the log queue on the Broker, you have to recreate the Broker Server storage.
 - Shut down the Broker Server;
 - Remove all `BrokerData.*` files from your Broker Server data directory;
Note: this will remove any transactions/documents that are currently in any client queues!
 - Restart the broker server.

Scenario 4: When there are documents left in Dead Letter Queue

When the Broker's dead letter client is configured, the Broker stores dead letters - documents for which the Broker has no subscribers - in the dead letter queue. By default, the client ID for the Broker's dead-letter queue is "DefaultDLQ_", however, you can optionally append a suffix to this client ID when you activate the queue. If there are dead letter documents in the dead letter queue, the dead letter client needs to retrieve the documents so they can be processed. Documents that remain in the dead letter queue occupy space in the Broker Server storage.

If this situation lasts for a longer time, the Broker Server will eventually run out of space. How quickly the Broker runs out of space depends on

- the size of the storage file(s);
- the size of the dead letter documents (see also [Calculating the size of a Broker document "on the wire"](#));
- the number of dead letter documents queued.

For more information about the dead letter queue, navigate to the Knowledge Base on Empower and search for the following article:

- 1614337494 (navigate to "Knowledge Center > Search The Knowledge Center" and enter this document ID in the "Search for:" "Any Document with ID" field)
Or refer to the Administering webMethods Broker Guide, chapter 5.

How to verify and what to check

- Check if there are any documents in the dead letter client's queue
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Clients and locate the client whose name has the "DefaultDLQ_" prefix. Check if there any documents in this queue by looking at the "Docs Queued" column in the client list.

Suggested Solution:

- Empty the dead letter queue
If there are documents in the queue of the "DefaultDLQ_" client, you need to clean up the queue by retrieving and processing the documents. How to handle the documents in the dead letter queue depends on your design. Dead letters are not processed by default, so if you configure the dead letter client, make sure you do not let dead letter documents pile up in the dead letter queue.

Scenario 5: When the Broker Server storage becomes fragmented

For performance reasons, the Broker's storage system divides each storage file into "regions", which are either Small (2MB) or Normal (8MB). Once a region has been allocated from a storage file, its type cannot change. Due to this, there are circumstances in which the storage system could fail making allocations. For example when there are not enough contiguous blocks available within the laid-out, empty regions to store an incoming document. In this situation the Broker will report "out of storage", even though the "Utilization" tab of the Administration > Messaging > Broker Servers > Broker Server Details page shows that there is enough free space.

How to verify and what to check

When the Broker Server is having storage problems due to fragmentation, check for the following symptoms.

- Check if the storage files can still grow
A way to indicate whether the Broker Server storage may be close to the risk of fragmentation, is to watch the "Current File Size" against the "Total Capacity" on the "Utilization" tab of the Broker Server Details page. Once "Current File Size" reaches 100% of the "Total Capacity", fragmentation may prevent publishing larger documents. This should therefore be monitored.
- Check if the storage is full
When you check the Broker Server's utilization, it indicates that the storage is not full; "Free" is more than 0 KB.
 - In My webMethods Server, navigate to Administration > Messaging > Broker Servers > Broker Server Details and click on the "Utilization" tab.
- Some documents can be published, but not all
When you publish documents to the Broker (each document with different size), some documents are published successfully and some fail with the error

"(100-2015): The operation failed because the Broker does not have enough storage to process the request. Alert your administrator"

Suggested solution

- Add more storage
The only solution to storage fragmentation issues is to increase the size of existing storage files or to add additional storage files (BrokerData.qs.stor). If you still encounter the same issue after increasing BrokerData.qs.stor, then you need to create a new BrokerData.qs.log with a bigger size. Currently there is no utility to defragment the Broker Server storage.

Important: adding more storage may resolve storage problems only temporarily

In the previous sections, many of the suggested solutions include adding more storage to the Broker Server. Note that this action will immediately free up the Broker to accept new requests, however it will not address the underlying cause.

Troubleshooting Tools

The following tools can help to obtain troubleshooting information.

Broker Storage Utility

The Broker Storage Utility is part of the Broker installation and can be used to troubleshoot issues related to the storage files of the Broker. For information about this utility, see the webMethods Broker Administrator's Guide.

Other Tools

QueueCheck.java

QueueCheck.java is a standalone Java program that uses a shared state client to allow you to print all the documents that are currently in a certain Broker client's queue and their data.

To compile QueueCheck.java, run the following command:

```
javac -classpath <webMethods_install_directory>/common/lib/wm-brokerclient.jar QueueCheck.java
```

For example:

```
javac -classpath C:\webMethods7\common\lib\wm-brokerclient.jar QueueCheck.java
```

Run the QueueCheck program using the following command:

```
java -classpath <webMethods_install_directory>/common/lib/wm-brokerclient.jar;. QueueCheck  
BrokerServer_hostname:BrokerServer_port BrokerName ClientGroup BrokerClientName
```

Where:

- **BrokerServer_hostname** is the machine name on which the Broker Server is running.
- **BrokerServer_port** is the port of that Broker server (default is 6849).
- **BrokerName** is the name of the Broker to which you want this QueueCheck program to connect, or the Broker name that has the Broker client that you use to check the documents in the queue (case sensitive).
- **ClientGroup** is the name of the Broker client group for which you want to check the documents in the queue (case sensitive).
- **BrokerClientName** is the name of the Broker client for which you want to check the documents in the queue (case sensitive).

Note: If you are getting a timeout error when running this java program, this indicates there is another session that is currently holding the document from this client queue. In this case, disconnect the other session before running this java program. For example, if this broker client is a trigger client, then you can shut down the Integration Server or disable the trigger client before running this java program.

AckFirst.java

AckFirst.java is standalone Java program that uses a shared state client to allow you to print the first document and acknowledge or remove it from a certain Broker client's queue.

To compile the AckFirst.java, run the following command:

```
javac -classpath <webMethods_install_directory>/common/lib/wm-brokerclient.jar AckFirst.java
```

For example:

```
javac -classpath C:\webMethods7\common\lib\wm-brokerclient.jar AckFirst.java
```

Run the AckFirst program using the following command:

```
java -classpath <webMethods_install_directory>/common/lib/wm-brokerclient.jar;. AckFirst  
BrokerServer_hostname:BrokerServer_port BrokerName ClientGroup BrokerClientName
```

Where:

- **BrokerServer_hostname** is the machine name on which the Broker Server is running.
- **BrokerServer_port** is the port of that Broker server (default is 6849).
- **BrokerName** is the name of the Broker to which you want this program to connect, or the Broker name that has the Broker client from which you want to obtain the first document (case sensitive).
- **ClientGroup** is the name of the Broker client group for which you want to obtain the first document out of its queue (case sensitive).
- **BrokerClientName** is the name of the Broker client for which you want to obtain the first document out of its queue (case sensitive).

Note: If you are getting a timeout error when running this java program, this indicates there is another session that is currently holding the document from this client queue. In this case, disconnect the other session before running this java program. For example, if this broker client is a trigger client, then you can shut down the Integration Server or disable the trigger client before running this java program.

ClearQueue.java

ClearQueue.java is a standalone Java program that enables you to delete the documents from the Broker client queue when you do not have access to the Broker Administrator. This is the same as clicking the **Clear** button on the Broker client.

To compile the ClearQueue.java, run the following command:

```
javac -classpath <webMethods_install_directory>/common/lib/wm-brokerclient.jar ClearQueue.java
```

For example:

```
javac -classpath C:\webMethods7\common\lib\wm-brokerclient.jar ClearQueue.java
```

Run the ClearQueue program using the following command:

```
java -classpath <webMethods_install_directory>/common/lib/wm-brokerclient.jar;. ClearQueue  
<BrokerServer_hostname>:<BrokerServer_port> <BrokerName> <ClientGroup> <BrokerClientName>
```

Where:

- **BrokerServer_hostname** is the machine name on which the Broker Server is running.
- **BrokerServer_port** is the port of that Broker server (default is 6849).
- **BrokerName** is the name of the Broker to which you want this program to connect, or the Broker name that has the Broker client from which you want to delete all the documents in its queue.
- **ClientGroup** is the name of the Broker client group for which you want to delete all the documents in its queue (case sensitive).
- **BrokerClientName** is the name of the Broker client for which you want to delete all the documents in its queue (case sensitive).

Note: If you are getting a timeout error when running this java program, this indicates there is another session that is currently holding the document from this client queue. In this case, disconnect the other session before running this java program. For example, if this broker client is a trigger client, then you can shut down the Integration Server or disable the trigger client before running this java program.

Common Symptoms and Issues

Broker Server Crashes (with Core Dump)

The Broker may crash for both *expected* and *unexpected* reasons. To troubleshoot Broker crash issues, see [Broker Crash Troubleshooting](#).

See the Unix Man page for "Signal (3HEAD)" for more information about the meaning of the signal that the Broker Server crashed with.

Broker Monitor Not Able to Start the Broker

If a Broker monitor cannot start the Broker, check to see if:

- That Broker monitor uses the same user ID and group as the command line.
- The filesystem rights are the same. The Broker monitor needs the same filesystem, memory, threads, and network access as the Broker.

Broker Logs Message That it Ran Out of Memory and Restarts

If the Broker reports to be out of memory, it means that the Broker tried to allocate some memory, but the OS did not allow this. At that moment, the Broker cannot continue normal execution, because some - possibly guaranteed - data cannot be stored. Therefore, the Broker exits immediately with exit code 1. This triggers the Broker Monitor to restart the Broker. Typically this causes minimal disruption of business processing.

Common causes for Brokers running out of memory are using large amounts of (large) documents that are all queued on the Broker at the same time, or using many different document sizes. That latter can cause heap fragmentation. On HP-UX, it's possible to mitigate heap fragmentation by setting an environment variable: `_M_ARENA_OPTS` to `1:8` or `1:32`.

Documents Stuck in Trigger Queue

Issues with documents stuck in the trigger queue can originate either on the Integration Server or on the Broker Server. Software AG recommends first examining the Integration Server and then examining the Broker.

Note: If resolving the issue is more important than troubleshooting it - for example, because of the way business is impacted by the issue - restarting the Integration Server and/or Broker Server will usually resolve the problem, at least temporarily. However, you should be aware that restarting the Broker takes away the opportunity to analyze what caused the issue. If you choose to restart the Broker Server, Software AG Global Support strongly recommends reproducing the issue in a non-production environment for further troubleshooting.

Examining the Integration Server

- Determine whether the problem is with one trigger or multiple triggers. Make a note of the trigger name. If multiple triggers are having the problem, start by focusing on one or two of them.
- Determine whether the trigger is set to Serial or Concurrent Processing. If it is set to Concurrent Processing, note the number of concurrent threads.
- Check the number of available threads in the Server Thread Pool and check the trigger throttle settings (**Settings > Resources > Trigger Throttle Controls**). If not enough threads are available, Integration Server will not process the documents.
- Determine whether the trigger is connected:
 - In the My webMethods Broker Administrator, click Sessions and note the Last Activity and Created times. Check to see if the Last Activity time gets updated.
 - Click the link next to **Documents in Queue** for the trigger client, and note the Last Queued and Last Retrieved times. They both should be more or less the same for a normal working client.
- If the Integration Server is not clustered, only one session should exist. If more than one session exists, this could indicate a stale session. Delete all sessions and restart Integration Server.
- Check the Integration Server server log and error log for errors related to Integration Server-to-Broker connections at that time.
- On the Integration Server Administrator Service Usage page, check if an instance of the trigger service is currently running. Note the Last Ran time.
- If you find that an instance of the trigger service is always running and the trigger is set to Serial Processing, obtain thread dumps of the Integration Server to determine what that trigger thread is doing or why it is stuck.
- If the trigger is set to Concurrent Processing with five threads, for example, and you see five instances of the trigger service running, Integration Server thread dumps will again help.
- See [Integration Server Pub-Sub Troubleshooting](#) for more information.

Examining the Broker

The problem is likely with the Broker if you find that:

- The trigger is connected.
- The sessions look okay.
- Sufficient Integration Server threads are available.
- The trigger service is not running or the Last Ran time is not close to the current time.
- There are no Integration Server-to-Broker connection errors.
- Restarting the Integration Server did not resolve the problem.

If the problem is happening to (almost) all of the triggers, it could be a Broker problem caused by a deadlock in Broker code. To troubleshoot Broker-side problems, gather the following information:

- Integration Server thread dumps
- Broker core dump (user dump on Windows)
- Broker version
- Operating system information
- CPU and memory used by Broker and Integration Server processes

Broker - Documents

Calculating the size of a Broker document "on the wire"

The envelope is about 200 bytes. Then the rest depends on your encoding. If you're sending "The quick brown fox jumps over the lazy dog" but send it in UnicodeString it uses 2 bytes per character. So large XML documents sent as unicode strings may be twice as large as you had expected. Of course, if they're full of non-ISO 8859-1 characters then they're closer to the original size, because that was already in double bytes.

If you know how you expect to be "encoding" your payload data then you can make estimates about size.

Number fields and structs in the document type matter too. If you're using many nested structs then there's some significant overhead because the name of each struct field that has a value gets transmitted.

The worst case is where you have a sequence of structs, such as:

```
struct {
  int int_value;
  float float_value;
  string string_value;
} myStructSeqField[];
```

If you transmit 100 struct values, then there are 100 copies of each of the 3 field names.

So if you're worried about that, you could change the encoding to:

```
int int_values[];
float float_values[];
string string_values[];
```

this would be harder to handle but there's only 1 copy of each field name.

The best way to know is to write a small custom client that fills out some example documents, calls "toBinData()" and reports the created size + 200 bytes.

The wire format is something like:

```
<FIELD_TYPE><FIELD_NAME><FIELD_VALUE>
```

Then there's some extra stuff for structs and sequences.

The format is binary in nature. Sending integer values always occupies 32bit regardless of actual value.

There's no conversion to an intermediate format. When you call

```
broker_event.setIntegerValue("foo", 42);
```

then the C API will generate the wire bytes and "append it" to the current document. The Java API actually stores it in a tree of values; when it's sent, the wireformat is generated.

For simple documents, assume: "200 bytes + sizeof(payload_data)*1.1". This is usually close enough. Unless it's the XML as unicode example.

Broker - Crash

Overview

The Broker automatically detects conditions that would cause data to be lost or permanently corrupted, such as inconsistencies in memory or the storage files. In order to preserve the data, the Broker actively causes a fatal runtime exception, which terminates the Broker Server process and triggers the operating system to generate a core dump. This is what is generally known as a Broker crash. After the crash, the Broker Monitor normally restarts the Broker Server to continue processing.

The core dump is a file that is located in the Broker Server's storage directory. It can be analyzed by Software AG for clues about the cause of the crash. Obtaining this file and providing it when reporting a support incident is critical in helping to resolve the problem.

This troubleshooting page provides information and a step-by-step process that applies to most Broker core dump support incidents. If you are running Broker on Windows, please also refer to the [Windows debugging](#) page.

The Broker Server Core File

On Windows, the Broker Server generates two files when it crashes:

- drwtsn32.log
- user.dmp

On UNIX and Linux, the Broker Server generates one file when it crashes:

- core

The following message in the Broker Server's log indicates that the Broker stopped unexpectedly:

```
1034 Unexpected stop of Broker Server in directory <broker_server_data_directory> pid
<awbrokerProcessId> Signal <SignalNumber> (core dumped)
```


Manually Generating a Core Dump on UNIX

You can manually cause a Broker core dump to be written by issuing any of the following commands:

- `kill -6 <process_id_of_awbroker>`
- `kill -ABRT <process_id_of_awbroker>`
- `gcore -o <core_file_name> <process_id_of_awbroker>`
The `gcore` command is part of the `gdb` utility on Red Hat, and the full development package may need to be installed on Red Hat to make those utilities available.
- `gencore <process_id_of_awbroker> <core_file_name>`
The `gencore` command generates a core dump without stopping the process on the AIX operating system.

Note: The "kill" commands will terminate the Broker Server process. "gcore" and "gencore" will generate a core dump while the Broker continues to run.

If you do not see the core file in the data directory of the Broker, issue the **ulimit** command to see if core files are allowed on your operating system. If your operating system does not allow core files, you will see command output similar to the following:

```
#ulimit -a
core file size (blocks, -c) 0
data seg size (kbytes, -d) unlimited
file size (blocks, -f) unlimited
max locked memory (kbytes, -l) 32
max memory size (kbytes, -m) unlimited
open files (-n) 1024
pipe size (512 bytes, -p) 8
stack size (kbytes, -s) unlimited
cpu time (seconds, -t) unlimited
max user processes (-u) 8190
virtual memory (kbytes, -v) unlimited
```

A core file size of 0 indicates that core files are not allowed to be written. To allow Broker to write core files, set the core file size to a value greater than 0. For example, to set the limit to 50000 bytes, issue the following command:

```
#ulimit -c 50000
```

Software AG Global Support recommends setting the `ulimit` value for core file size to "unlimited". This is to avoid a core file being 'truncated' if the crashing Broker's memory footprint exceeds the preconfigured `ulimit` value. Truncated core files generally do not provide sufficient information to determine the cause of the Broker crash.

You can use a debugging tool such as **dbx** (on Solaris and AIX) or **gdb** (on HP-UX and Linux) to debug the core file.

Manually Generating a Core Dump on Windows

On Windows, you can use the tool called 'userdump.exe' to generate a core dump file of the broker. From the command line, issue the following command:

```
userdump <process_id_of_awbroker>
```

The userdump tool is not part of a Windows installation by default. It can be downloaded from Microsoft. For more information and access to the download location of userdump, please refer to <http://support.microsoft.com/default.aspx?scid=kb;en-us;241215>

Automatic Broker Restart

If the Broker Server process terminates, the Broker Monitor will try to restart the Broker Server automatically, up to three times, unless it is configured not to. If the Broker Monitor does not restart the Broker Server within three attempts, you must start the server manually. Whether the Broker should be automatically restarted is configured in its configuration file: "awbroker.cfg", line

```
auto-restart=0
```

Note: if this line is not present, or has any value other than 0, the Broker is automatically restarted when it exits with a non-zero exit code.

Determining the Cause of a Crash

To help Software AG Global Support determine the cause of the crash, you must first collect crash data.

Collect Crash Data Using a Script

- **Obtain the Broker core collect script from Software AG Global Support.**

This script automatically gathers all required data into one zip file.

The format of this script is as follows:

```
broker-core-collect.sh [-zZGnrdsHV] [-e <exe> ] [-c <core>] <datadir>
```

- z: use zip to package instead of tar
- Z: compress tar file (assumes compress in PATH)
- G: gzip tar file (assumes gzip in PATH)
- n: do not archive collection (implies -r)
- r: retain temporary files
- d: include data/storage files
- s: silent, no verbose output, only errors
- h: online help
- v: print version
- e exe: override location of Broker Server executable
- c core: corefile to package (otherwise copies all cores in datadir)

Notes:

- The parameter <datadir> is required.
 - Temporary files are placed in \$TMP (or /tmp if \$TMP is not set).
 - The final archive is placed in the current directory.
- Software AG Global Support will guide you on which parameters to specify for your situation. The script also includes additional instructions for running the script.

- **Transfer the file wmbroker-crash-<yymmdd>-<HHMMSS>.<archive extension> via FTP**

Prerequisites

The following prerequisites must be met to obtain full troubleshooting information.

Enable Full Core Dumps on AIX

NOTE: AIX users must enable full core dumps as described in the AIX documentation (http://www-128.ibm.com/developerworks/eserver/library/es-javaonaix_trace.html).

To enable full core dumps for the system, run the following root user command:

```
chdev -l sys0 -a fullcore=true
```

This change does not require a system reboot. If you are familiar with the SMIT utility, you can change the setting by running the command **smitty chgsys** and setting the value for **Enable full CORE dumps** to true.

Broker - Territory

Overview

Brokers can be linked to form units known as territories. Brokers that form a territory function as a single logical Broker. Clients that connect to one Broker in a territory can automatically exchange documents with clients connected to any of the Brokers in the territory. Territories support scalability by allowing you to segment traffic among multiple Brokers, while still treating the Brokers as a single operational unit.

This troubleshooting page provides information to help troubleshoot territory issues.

Obtaining Territory Statistics

This section describes a sample Java program, TerritoryStats, that allows you to connect to one of the brokers in the territory to obtain the territory statistics of another broker(s) in the territory.

For example: You have two brokers (Broker1 and Broker2) in the territory called ProductionTerritory.

- To obtain the territory statistic information about Broker1, you run this TerritoryStats program to connect to Broker2.
- To obtain the territory statistic information about Broker2, you run this TerritoryStats program to connect to Broker1.

The output of TerritoryStats will look something like this:

```
Below are the territory statistics for:
brokerHost = system.abc.com
brokerName = Broker2
event struct_value {
unicode_string brokerHost = "system.abc.com";
unicode_string brokerName = "Broker2";
unicode_string description = "";
unicode_string connectedFrom = "192.168.1.100";
date lastConnectTime = "03/23/2010 22:05:50";
int numEventsForwarded = 1;
int numEventsReceived = 1;
date lastEventForwardTime = "03/23/2010 22:05:50";
date lastEventReceiveTime = "03/23/2010 22:05:50";
int numEventsEnqueued = 1;
date lastEventEnqueueTime = "03/23/2010 22:05:49";
long queueLength = 0;
long queueByteSize = 0;
long queueHighestLength = 1;
date queueHighestLengthTime = "03/23/2010 22:05:49";
string protocol_version = "6.6";
};
```

1. Save the following source code as TerritoryStats.java.

```
import COM.activesw.api.client.*;
import java.io.*;

public class TerritoryStats
{
    static String brokerHost;
    static String brokerName;

    public static void main(String[] args)
    {
        String brokerhost;
        if(args.length != 2){
            System.out.println("Provide the name of <BrokerServer[:port]> <Broker name>");
            System.exit(0);
        }
        brokerHost = args[0];
        brokerName = args[1];
        BrokerAdminClient c;
        BrokerEvent getStats;

        try {
            c = new BrokerAdminClient(brokerHost, brokerName, null, "admin", "getStats", null);
            getStats = c.getTerritoryStats();
            BrokerEvent brokers [] = getStats.getStructSeqFieldAsEvents("brokers", 0, -1);

            for(int i=0; i < brokers.length; i++) {
                System.out.println("Below are the territory statistics for:");
                String brokerHost = brokers[i].getStringField("brokerHost");
                String brokerName= brokers[i].getStringField("brokerName");
                System.out.println("brokerHost = " + brokerHost );
                System.out.println("brokerName = " + brokerName);

                System.out.println(brokers[i].toString());
            }

            // disconnect
            c.destroy();
        }
        catch(BrokerException e) {
            System.out.println("Error creating BrokerHostClient: " + e);
        }
    }
}
```

Use the following command to compile the Territorystats.java file from the directory where TerritoryStats.java is located:

```
javac -classpath <webMethods_install_directory>/common/lib/wm-brokerclient.jar TerritoryStats.java
```

For example:

```
javac -classpath C:\webMethods7\common\lib\wm-brokerclient.jar TerritoryStats.java
```

1. After compiling `TerritoryStats.java`, it will create this class file, `TerritoryStats.class`, at the same directory where `TerritoryStat.java` is located.
2. From the directory where `TerritoryStats.class` is located, execute it using the below command (execute as one command):

```
java -classpath <webMethods_install_directory>/common/lib/wm-  
brokerclient.jar;<webMethods_install_directory>/common/lib/wm-g11nutils.jar;.  
TerritoryStats <Broker_Server_hostname>:<Broker_Server_port> <Broker_Name>
```

Where

- `<Broker_Server_hostname>` is the machine name on which the Broker Server is running.
 - `<Broker_Server_port>` is the port of the Broker Server (default is 6849).
 - `<Broker_Name>` is the name of the Broker to which you want this program to connect.
- For example:

```
java -classpath C:\webMethods7\common\lib\wm-  
brokerclient.jar;C:\webMethods7\common\lib\wm-g11nutils.jar;. TerritoryStats  
system.abc.com:6849 Broker1
```

Adapters - JDBC Adapter

Diagnostics

This section lists the diagnostic information that may be obtained for troubleshooting JDBC Adapter.

General Information about JDBC Adapter Logging

- All messages are logged through the WmART logger
- The major facility code is 1
- Information messages are logged at Trace level
- Exceptions are passed on to WmART as an AdapterException, AdapterServiceException (for the adapter services), or AdapterConnectionException. An SQLException (from the RDBMS) is wrapped into an AdapterException.

The JDBC Adapter's log message appears in one of the following formats:

```
ADA.1.nnnnc
ADA.0001.nnnnc
ADA stands for Adapter and '1' stands for JDBC Adapter
```

JDBC Adapter has a dependency on the WmART package. Ensure that the following facility is enabled:

- 0113 Adapter Run time (Managed Object)
- 0114 Adapter Run time
- 0115 Adapter Run time (Listener)
- 0116 Adapter Run time (Notification)
- 0117 Adapter Run time (Adapter Service)
- 0118 Adapter Run time (Connection)
- 0121 Adapter Run time (SCC Transaction Manager)
- 0126 Adapter Run time (SCC Connection Manager)

JDBC Adapter Logging (JDBCLogFile)

Use this option to enable tracing at driver level. Most JDBC drivers support logging at driver level. JDBC Adapter Logging reports very low level adapter logging, including the SQL statements passed to the database driver. To enable JDBC Adapter Logging, add the following property either directly into server.cnf, or into the extended properties:

```
watt.adapter.JDBC.JDBCLogFile=jdbc.log
```

Note: For Oracle you need to use the driver with _g extension to enable driver logging. (Example: *_g.jar which includes debugging information and java.util.logging calls. For more details about how to enable Oracle driver logging, check the following link : [oracleDriverFaq](#)):

JDBC Connection Properties

You can specify extra properties in the JDBC Connection **Other Properties** field. These settings are specific to the database driver in use.

Example (applicable to the Oracle driver):

```
Other Properties: driverType=thin;connectionProperties={oracle.jdbc.ReadTimeout=60000}
```

For more information about available properties for the database driver you're using, refer to the documentation supplied by its vendor.

Common Problems

Adapter Exception Handling

JDBC Adapter throws two kinds of exceptions:

AdapterException

JDBC Adapter throws an AdapterException for two reasons:

- To report a configuration error or a connection creation error
- To wrap an SQLException (thrown from a database) if the adapter does not consider the SQLCODE to be a fatal error. In this case, WmART wraps the AdapterException in a com.wm.pkg.art.error.DetailedServiceException and throws it to Integration Server. AdapterExceptions containing an error code of 316 are SQLExceptions.

AdapterConnectionException

JDBC Adapter throws an AdapterConnection exception if the adapter interprets the SQLCode as a fatal error. If a fatal error is encountered, the following will be observed:

- WmART 6.5 and later reset the entire connection pool (WmART 7.x.x and 8.x)
- WmART earlier than 6.5 drop the connection from the connection pool

WmART then wraps the exception in a com.wm.pkg.art.error.DetailedSystemException and throws it to the Integration Server.

SQLException

JDBC Adapter throws an SQLException if the JDBC driver fails to execute an SQL Command against a database. Typically, the exception will contain the SQL STATE, an SQLCODE, and an error message. (Most likely the error occurs because of a problem on the database side or executing the wrong SQL against the database. Based on the SQL State and SQLCode, we can establish the reason for the error from the database documentation.)

Fatal Error Codes

If you receive a specific database error code, you can configure JDBC Adapter to automatically refresh the JDBC Adapter connection. To trigger the JDBC Adapter to refresh the connection upon receiving a specific error, set the error code in the list of fatal error codes. Specify the property in the Integration Server extended settings as follows:

```
watt.adapter.JDBC.<database driver>.fatalErrors=+ErrorCode_1,ErrorCode_2,ErrorCode_n
```

The watt property is database driver specific. Replace *database driver* with the name of the respective database driver. Note that there is no space after the ',' (comma).

The following example is for Oracle JDBC driver to refresh a connection when error codes 17002 and 17003 are encountered:

```
watt.adapter.JDBC.Oracle.fatalErrors=+17002,17003
```

Connection Not Available

This issue occurs when all the JDBC Adapter connections in a pool (with pool size set to max connections) are used by the JDBC Adapter services. You may notice the following error in the Integration Server log file:

```
[SCC.0126.0106E] An error occurred while attempting to retrieve a connection from <...>
[ART.117.4012] Adapter Runtime (Adapter Service): Unable to run adapter service. Error occurred
when connecting to resource <...>.
```

or

```
[ART.118.5053] Adapter Runtime (Connection): Unable to get a connection to resource <...>
A connection was not available for request in pool <...>
```

In most cases, this issue can happen due to high load when there are simply not enough connections available in the pool to process the request. You can try to increase the max pool size. However, if the issue persists even after increasing the pool size, possible causes are:

- The database does not release the connection. Response: find out the cause for this behavior. Other applications might be causing the lock on the database.

- A connection is not released to the pool. Response: obtain thread dumps on Integration Server to verify this behavior. Find out if any services are still running (that is, the connection uses the adapter services). If the connection is transaction based, it is possible that the transaction is not committed.

JDBC Adapter Page Shows Up Blank

1. Check the Service Usage page to see if there are any hanging services that invoke JDBC transactions.
2. Check if there are any database issues.
3. Check the Server.log and Error.log
4. Run WmART/pub.art.connection:getConnectionStatistics with the JDBCConnection name as the alias name and capture the output.
5. Check the database to see how many connections are open from Integration Server for that user.
6. Obtain 3 thread dumps of Integration Server allowing an interval of 30-60secs between each two.
7. Check for threads from the adapter code or driver code doing socketRead or SocketConnect. Check also if the state remains unchanged in all three thread dumps.
8. Check for any deadlocks on the database server using a database administration client.
9. If the JDBC Adapter threads hang during socketRead or SocketConnect, set QueryTimeout or LoginTimeout respectively in the **Other Properties** field of the adapter connection.
10. If you still see threads in socketRead or socketConnect originating from the driver code, this indicates that the driver code does not handle any network problems that may be occurring.

JDBC Adapter Notifications: Records Not Being Picked Up

1. Ensure that the adapter notification is enabled from the JDBC Adapter page.
2. Verify that you have a service that deletes records from the buffer table after the records have been processed.
3. If the notification is clustered, un-cluster it to determine if the problem still occurs. This will help isolate the problem. If the problem does not occur when the notifications are non-clustered then the problem may be in the repository. Follow the troubleshooting steps for repository related problems in this guide.
4. Check if any JDBC Adapter errors and scheduler-related errors exist in the Integration Server logs.
5. Obtain three thread dumps at an interval of 30-60secs. Look for hanging threads, if any.

Error: "pool doesn't have connections available"

1. Check the pool settings from the JDBC Adapter configuration page. Check if the maximum connection pool size is sufficient for the load on your system.
2. Run the pub.art.connection:getConnectionStatistics service from the WmART package to confirm that all connections are busy.
3. Install the latest fixes for the WmART package.
4. After performing the other steps on this list, capture the logs at log level 8 and obtain three thread dumps at an interval of 30-60secs.

Database Specific Issues

Connecting to Oracle RAC/OCI Driver

OCI Driver

1. Ensure that you have installed the Oracle client libraries on the host where JDBC adapter is installed.
2. Ensure that the Oracle client libraries are compatible with the database server to which the JDBC Adapter is connecting.
3. Ensure the client libraries are included in the "PATH" used by JDBC Adapter.
4. Verify that you have set "driverType=oci" in the **Other Properties** of the JDBC Adapter connection.
5. If the other steps on this list do not resolve the issue, test the connection using a simple java code outside of webMethods Integration Server, preferably using the same driver, datasource and URL, and from the same box.

Oracle RAC

- JDBC Adapter can be configured to connect to RAC using a thin driver or OCI driver.
- Following is a Sample URL for a thin driver:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=
(ADDRESS=(PROTOCOL=TCP)(HOST=node1)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=node2)(PORT=1521))
(LOAD_BALANCE=yes)(FAILOVER=on))
(CONNECT_DATA=(SERVICE_NAME=rac_db)))
```

You can set the URL in the **Other Properties** field of your JDBC connection:

```
driverType=thin;url=jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on)
(ADDRESS=(PROTOCOL=TCP)(HOST=host1)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=service)))
```

Connecting to DB2

- We support DB2 on AS400/OS390/UDB.
- IBM now supports a type-4 driver for all three databases. Whenever possible, use type-4 driver as opposed to type2/3.

Problem Configuring JDBC Adapter to Connect to Oracle RAC/OCI driver/DB2

Invalid <database> URL specified: <DataSource>.makeURL

1. Ensure that the specified datasource class and URL are valid.
2. Verify that the parameters in the URL, such as port and databaseName, are valid.
3. Verify that the existing properties in the **Other Properties** field are valid.

Error Establishing Socket, Network Adapter Not Found, or Connection Refused

The following items may help resolve these errors:

1. If it is a Type-2 or Type-3 driver, ensure that the client installation and the communication channel all the way to the server is good.
2. If it is a Type-4 driver, ensure that there are no network issues between the server where JDBC Adapter is installed and the server where the database is hosted.
3. Check if there is a firewall blocking the connection between Integration Server and the database server.
4. Ensure that the database server is running on the host and port you specified. Run telnet or ping command to test it.
5. Test the connection using a simple java code outside of webMethods Integration Server, preferably using the same driver, datasource, and URL.

Known Issues with Configuring Oracle RAC/OCI driver/DB2

No Tables Listed for AdapterService

- Check if the user has permissions to access all required tables.
- Check the **Other Properties** field in the adapter connection to ensure that there is no filter restricting access to schemas.
- Write a standalone java client or a java service in Integration Server using the following methods to see if the list of tables can be retrieved using the same user and connection details:

```
...
DatabaseMetaData meta = conn.getMetaData()
ResultSet rs = meta.getSchemas()
ResultSet rs = meta.getTables()
...
```

No Fields listed for Adapter Services

- Check if the database object used in the adapter service is a 'table' or a 'synonym' (for Oracle). If it is a 'synonym' (for Oracle), this is a known issue for a number of drivers.
- If you are using one of the latest drivers, try setting this property in the **Other Properties** field of the adapter connection as follows:

```
includeSynonyms=true
```

- Database SYNONYM names and SYNONYM column information are generally retrieved by the JDBC DatabaseMetaData getTables and getColumns methods respectively. We code the adapter using those general methods. Experience with JDBC drivers shows that the two DatabaseMetaData methods do not always work. You can work around SYNONYM issues by using a custom SQL template, dynamic SQL template, or even write a stored procedure to achieve the same effect as using the SELECT template.
- To test this outside of webMethods Integration Server:

```
...
DatabaseMetaData meta = conn.getMetaData()
ResultSet rs = meta.getSchemas()
ResultSet rs = meta.getTables()
ResultSet rs = meta.getColumns()
...
```

Checklist Before Opening a Support Incident

This checklist includes any action items that should be carried out by the customer prior to opening a Support Incident.

- Ensure that both JDBC Adapter and WmART packages are loaded properly.
- If there is a problem with connecting or executing an adapter service, ensure the connection or execution of statement works fine outside of Integration Server. For example, test using java code. Also, try with the latest version of the driver available from the database vendor.
- Check if the WmART package and JDBC Adapter are installed with the latest fixes.
- If there is a service running long or a hang situation, obtain three thread dumps on Integration Server.
- Ensure that you use the correct database drivers and libraries.

Information to Attach to the Support Incident

- Integration Server About Page and Updates Page (<http://<host>:<port>/WmRoot/updates.dsp>)
- Fixes applied to WmJDBC and WmART
- Database vendor, exact version, patch level and operating system
- Which version/type of JDBC driver is used
- Integration Server Cluster settings
- Add the complete error related to JDBC Adapter or the WmART package (including the stack trace), if any exist in the Intergration Server Error log
- The JDBC Adapter connection properties
- WmART/pub.art.connection:getConnectionStatistics information for connection related issues.

Adapters - SAP Adapter

Introduction

webMethods SAP Adapter provides functionality for integrating an SAP system into the webMethods platform. SAP Adapter allows you to extend your SAP business processes and integrate non-SAP products using open and non-proprietary technology. SAP Adapter allows for both asynchronous and bi-directional, real-time communication to and from the SAP system.

Common Problems and Typical Causes

Integration Server Process Crashes When Processing Large IDocs

Common Symptoms

SAP Adapter will cause Integration Server to crash or terminate abruptly in one of the following scenarios:

- A single IDoc with a large number of segments is sent from SAP to Integration Server
For example: a single IDoc with over 19,000 segments.
- A batch of IDocs with a large numbers of IDocs is sent from SAP to Integration Server
For example: a batch of IDocs with around 2200 IDocs, where each IDoc contains around 24 segments.

After Integration Server terminates, you will get one of the following error messages based on the operating system on which you run Integration Server:

- On a Windows platform, you see an error in the Event Viewer log with the Description message: "The webMethods Integration Server service terminated unexpectedly".
- On a Unix platform, you see a message in the system log indicating that the Java application has terminated. If the system is configured to generate a core dump, you will find a file named "core" in the SoftwareAG_install_directory/IntegrationServer directory.

Possible Causes

This is a well known issue with the SAP RFC library. The RFC library allocates memory dynamically from the system heap while processing IDocs. When there is no sufficient memory left in the system heap, the RFC Library will exit without warning or error. This terminates the hosting process, which in this case is the Java process running Integration Server.

Possible Solution and Recommendations

- For information about how to handle large IDocs, see the following Technical Article on the Empower web site. ([Empower KB Article WSP - 1711708](#):How to handle memory problems and crashes when transmitting large documents)
- For SAP Adapter version 6.5, we recommend that you upgrade the SAP libraries. In recent versions of the SAP libraries SAP has addressed some of the memory issues. The following versions of the RFC libraries consume less memory and that may help in reducing the size of the Java system heap:
 - JCO version 2.1.8
 - jRFC version 2.1.8
 - IDoc library version 1.0.8
- To alleviate memory limitations on the operating system level, upgrade your 32-bit operating system to a 64-bit operating system.
- If none of the above options applies, split your IDoc (or batch) into multiple smaller IDocs (batches) on the SAP side before sending them to Integration Server.

Integration Server Crashes with JCO Library Version 2.1.7

It has been reported on several occasions that JVM crashes when using version 2.1.7 of the SAP libraries. SAP discovered a problem with this library version and you are strongly advised not to use it. You must either roll back to 2.1.6 or upgrade to 2.1.8.

SAP User Name/Password Invalid

If you run into error setting up the User Name and Password in the SAP Adapter configuration screen, make sure you use only capital letters for the User Name value. The SAP system requires the use of all capitals for the User Name value when specifying a User Name and Password.

SAP Adapter 6.5 Causes Very High CPU Load, Even When Idle

The Routing Listener causes high CPU utilization on idle systems. This issue was resolved by Fix 1 for SAP Adapter 6.5. This fix is included in any subsequent fixes. Download the latest superseding fix for SAP Adapter 6.5 from the Empower web site.

Thread Synchronization and Locked Sessions

Threads must never use the same RFC connection concurrently. If the same connection is used in multiple threads simultaneously, you will see this JCO error in the Error log:

```
com.sap.mw.jco.JCO$Exception: (132) JCO_ERROR_CONCURRENT_CALL: Concurrent call. Connection currently used in another thread
```

To resolve this issue, ensure that:

- Each "lockSession" invocation is paired with a "releaseSession". A service must never exit in-between these two calls.
- A section enclosed by "lockSession" and "releaseSession" is never executed concurrently by two or more threads.

Issues with SAP RFC Libraries

If issues occur because of problems in the SAP RFC libraries, the issues may be resolved by upgrading to the latest available version of the RFC libraries.

Minimizing the Number of Segments in IDoc

Issues related to performance and resources (for example, memory usage) often occur because of the size of an IDoc that is being processed. You need to establish:

- How many segments there are in the IDocs being processed.
- How many of these IDocs are being processed concurrently.

For information about the correlation between memory and IDoc segments, see the chapter on system requirements in the webMethods SAP Adapter User's Guide.

Checklist Before Opening a Support Incident

The following checklist specifies action items that you must complete before you open a Support Incident:

- Ensure that any third party libraries have the correct versions.
- Ensure that the correct Support Packs for any third party software are installed.
- Ensure that the latest third party fixes were installed in the correct order.
- Ask your SAP experts to look into the issue.

A good number of issues could be traced back to an issue on the SAP side, even when they were first discovered in webMethods SAP Adapter.

- Go through the list of [Common Problems and Typical Causes](#).

Information to Gather When Opening a Support Incident

SAP Adapter About Page

This page contains the following information:

- SAP Adapter Version, SP-Level and Build
- JCo Version
- JCo Middleware
- JCo Middleware Version
- jRFC Library Version
- RFC Library Version
- jARM Version
- OS Version

Log files

For SAP adapter version 6.5, the logging output from SAP Adapter is logged into the SAP <timestamp>.log file, located in the "<SoftwareAG_install_directory>/IntegrationServer/WmSAP/logs" directory.

For SAP Adapter version 7.1 and later, the logging output from the adapter is logged into the Integration Server server.log file, located in the "<SoftwareAG_install_directory>/IntegrationServer/logs" directory.

If you can reproduce the error, you must increase the SAP logging level setting to 10, recreate the problem, and send the log files to Software AG Global Support.

For information about how to set or change the SAP Adapter logging level, see the chapter on logging and monitoring in the webMethods SAP Adapter User's Guide.

RFC Trace Logs

If the issue is related to the function modules that you call or use, we recommend that you enable the RFC trace for the SAP Adapter listener and SAP Adapter connection. Then you can further analyze the data recorded in the RFC trace files: `rfc*.trc` and `dev_rfc.trc`.

For more information about how to enable or monitor the RFC trace files, see the webMethods SAP Adapter User's Guide.

Connection Tests

Perform the following tests to determine connectivity:

- SAP -> Integration Server (inbound)
 - Verify that the SAP listeners in the SAP Adapter are enabled.
 - Test the connection from SAP to Integration Server using transaction "SM59".
 - Verify that the network is functional. Use a ping program to "ping" the Integration Server host from the SAP Gateway service host.
 - Examine the SAP Adapter's RFC trace files for errors.
- Integration Server -> SAP (outbound)
 - Verify that the network is functional. Use a ping program to "ping" the Integration Server host from the SAP Gateway service host.
 - Verify that your user on Integration Server has the appropriate access permissions to the SAP system.
 - Increase the SAP Adapter logging level to 10 and perform the "Test Connection" operation from SAP Adapter. Check for any connection-related error messages in the log.

JVM Crash Reports and Core Files

When the JVM crashes, follow these steps to troubleshoot this issue:

1. Check if any of the following files are present in the `IntegrationServer_directory` or any of its subdirectories:
 - `hs_err*`
Example: `hs_err_pid1234.log`
 - `javacore*`
 - `java_core*`
 - `core`
2. If you find any of these files, check the timestamp of the file to determine if it was generated at the time of the JVM crash.
3. Send the file to Software AG Global Support.

Adapters - Siebel Adapter

Introduction

When installing webMethods Siebel Adapter, you will notice in the Software AG Installer that the Siebel Adapter consists of the following distinct components:

- **webMethods Siebel Adapter 6.0/Program Files**
- This is the Integration Server-side component that resides on Integration Server and provides webMethods-to-Siebel communication.
- After installing this component, you can access:
 - The WmSiebelAdapter package located under the `<SoftwareAG_install_directory>/<IntegrationServer_directory>/packages/` directory. To view or manage the package, go to "Integration Server Administrator > Packages > Package Management" page.
 - A new user interface for the Siebel Adapter on the "Integration Server Administrator > Adapters > Siebel Adapter" page.
- **webMethods Siebel Adapter 6.0/Siebel Server Component**
- This is the Siebel Server-side component (called EAI webMethods Transports) that provides Siebel-to-webMethods communication.
- After installing this component, the EAI webMethods transport files (`WmISTransportAix.zip`, `WmISTransportHpux.zip`, `WmISTransportSol.zip`, and `WmISTransportWin.zip`) are installed. The files are initially placed in your `"<SoftwareAG_install_directory>/<IntegrationServer_directory>/packages/WmSiebelAdapter/backend"` directory, and must be manually moved to the Siebel Server system. For details about installing the adapter, see the *"Siebel Adapter Installation Guide"*.

Common Problems

Siebel Adapter Fails to Start a Siebel Java Connection

You may see the following error message:

```
[...]threw a SiebelException: code(8239), msg(NSReadKey request failed (no error information).)
```

If the Siebel Java Connection (that is, the Siebel Adapter connection) fails to start, perform the following check:

1. Verify that the Siebel Java Data Bean .jar files (Siebel.jar and SiebelJI_enu.jar) are in the correct location on the Integration Server file system. The files should be located here:
"<SoftwareAG_install_directory>/<IntegrationServer_directory>/packages/WmSiebelAdapter/code/jars" directory.

These .jar files come from the Siebel Server (they do not ship with any Software AG software). These .jar files must be copied from the Siebel Server and placed in the WmSiebelAdapter\code\jars directory. This step requires restarting Integration Server.

2. Verify that Integration Server uses the correct version of the Siebel Java Data Bean jar files (SiebelJI_Common.jar, Siebel.jar, and SiebelJI_enu.jar). For example, if you are connecting to Siebel Server 7.5.2, do not use the jar files from Siebel Server 7.0.4. Note that different Siebel Server versions may package the necessary libraries differently and use different .jar file names.
3. Verify the following connection parameters for the Siebel Server (preferably with the Siebel system administrator):

```
Gateway server : The name of the Siebel Server Gateway.
Siebel Enterprise : The name of the Siebel Enterprise.
Object Manager : The name of the Siebel Object Manager.
Siebel Server : The name of the Siebel Server.
Username/Password : username/password to log in to the Siebel Server.
Repository Context : The Siebel repository in which the Siebel application stores the
business component and business object definitions.
By default, the name of the Repository Context is "Siebel Repository".
```

4. If you are still unable to start an adapter connection, you can use the stand-alone Java program to test the connectivity to your Siebel Server. This Java program only uses Siebel's Java API and has no webMethods code in it. This will further validate the connection parameters being used. If this client works, but Siebel Adapter still has issues, open a Support Incident with Software AG Global Support for further assistance.

Siebel Adapter Fails to Start a Siebel Windows Thin Connection

If the Siebel Thin Connection fails to start, perform the following check:

1. You need to have Siebel Windows Thin client or Siebel's COM Data Control installed on the same box where the Siebel Adapter is installed.
2. Verify the following connection parameters for the Siebel Server (preferably with the Siebel system administrator):

```
Gateway server : The name of the Siebel Server Gateway.
Siebel Enterprise : The name of the Siebel Enterprise.
Object Manager : The name of the Siebel Object Manager.
Siebel Server : The name of the Siebel Server.
Username/Password : Username/password to log on to the Siebel Server.
Repository Context : The Siebel repository in which the Siebel application stores the
business component and business object definitions.
By default, the name of the Repository Context is "Siebel Repository".
```

3. If you are still unable to start an adapter connection, you can use the stand-alone Visual Basic sample to test the connectivity to your Siebel Server. This Visual Basic program has been built using Siebel COM API and does not have any webMethods code in it. This will further validate the connection parameters being used. If this client works, but Siebel Adapter still has issues, open a Support Incident with Software AG Global Support for further assistance.

Unable to Invoke Methods from EAI webMethods Transports

If you receive the following error while attempting to invoke any methods from the EAI webMethods Transport package, this indicates that the EAI webMethods Transports were not installed or configured properly.

```
*****Error Message Start*****
SystemError 6003: Cannot load dll module "wmISTransport" Error code: 670884880.Error near no
filename:820 [dynamicLink()]
*****Error Message End*****
```

Verify the following on your Siebel Server file system:

- The wmISTransport.dll and wmTransport folders exist in the <SiebelServer_install_directory>\siebsrvr\bin directory.
- The WmISTransport.jar file is located in the <Siebel_install_directory>\siebsrvr\bin\wmTransport directory.
- The client.jar file (for Integration Server 7.0 and earlier) or the wm-isclient.jar file (for Integration Server 7.1 and later) are located in the <SiebelServer_install_directory>\siebsrvr\bin\wmTransport.
- The PATH system environment variable must be set to one of the JDK directories where the jvm.dll file is located, for example:
- (for JDK 1.5.x) PATH=c:\jdk1.5.x\jre\bin\client;c:\jdk1.5.x\bin;c:\jdk1.5.x\jre\bin;%PATH
- You must have a system environment variable called "WEBMTCFG" pointing to the directory where the wmISTransport.jar and wmISTransport.cfg files are located. It should typically point to the <SiebelServer_install_directory>\siebsrvr\bin\wmTransport directory.
- The CLASSPATH system environment variable must also be setup correctly to include various libraries (including the wmISTransport.jar file).

Many of these changes will require restarting the Siebel Server. For more information, see the Siebel Adapter Installation Guide.

Field Names With Special Characters

If you see the following error:

```
Field "Input Field" has illegal characters in its value "Exclude R/O at Cashout". An input or output field may not contain (,), [,], ,/,, ,i, or %.
```

This error occurs when a service uses reserved characters, such as '%', '/', etc. which Integration Server does not allow in generated service input or output names. The workaround is to modify the field names of the Siebel Server not to use such characters.

Information to Gather When Opening a Support Incident

- Environment Details About Where the Problem Occurs
- List of Installed Fixes
- Log file Associated with the Component Facing the Issue

Adapters - Siebel Connection Sample Code

Standalone Java Program Code Sample

This standalone Java program has been built using Siebel Java API. To run the Java program, follow these steps:

This standalone Java program has been built using Siebel Java API. To run the Java program, follow these steps:

1. Save the following Java code in a file and name the file "DataBeanDemo.java".

```

-----Sample Program Start-----
import com.siebel.data.*;
import com.siebel.data.SiebelException;

public class DataBeanDemo
{
    private SiebelDataBean m_dataBean = null;
    private SiebelBusObject m_busObject = null;
    private SiebelBusComp m_busComp = null;
    public static void main(String[] args)
    {
        DataBeanDemo demo = new DataBeanDemo();
    }
    public DataBeanDemo()
    {
        try
        {
            // ##### change me - begin #####
            String connect_str = "siebel.tcpip.none.none://gatewayserver/enterpriseServer/ObjMgr/SiebelServer";
            // For example: to connect to mySiebelServer1 siebel server.
            //String connect_str
            //="siebel.tcpip.none.none://kayak1.west.webmethods.com/siebel/SCC//ObjMgr_enu/mySiebelServer1";
            // Kayak1 is a Windows 2000 Server box, running Siebel 7.5.2 for Oracle 8.1.7.

            String user_id = "sadmin";
            String password = "sadmin";
            String lang = "ENU";
            // ##### change me - end #####

            // instantiate the Siebel Data Bean
            m_dataBean = new SiebelDataBean();

            // login to the server
            System.out.println("Logging into Siebel using...");
            System.out.println("connect_str -> " + connect_str);
            System.out.println("user_id -> " + user_id);
            System.out.println("password -> " + password );
            System.out.println("lang -> " + lang );

            boolean status = false;
            try
            {
                status = m_dataBean.login( connect_str, user_id, password, lang );
            }
            catch (SiebelException e)
            {
                System.out.println("Error Code->" + e.getErrorCode());
                System.out.println("Error Message->" + e.getErrorMessage());
            }
            System.out.println("Login status..." + status );

            if (status)
                System.out.println("Logged into Siebel as..." + m_dataBean.loginName());
            else
                System.out.println("Login to Siebel failed");
        }
    }
}

```

```
// get the business object
//m_busObject = m_dataBean.getBusObject("Opportunity");

// get the business component
//m_busComp = m_busObject.getBusComp("Opportunity");

// logoff
m_dataBean.logoff();
}
catch (SiebelException e)
{
System.out.println("Error Code->" + e.getErrorCode());
System.out.println("Error Message->" + e.getErrorMessage());
}
}
}

-----Sample Program End-----
```

2. Open the DataBeanDemo.java file and modify the section between the "change me - begin" and "change me - end" comments to include the connection parameters of your Siebel Server.
3. Save the file.
4. Compile the java code from a command prompt using:

```
javac -classpath <full_path_of_Siebel_databean_jar_files> DataBeanDemo.java
```

For example, if connecting to a Siebel Server 7.5.2, run the command with SiebelJI_Common.jar and SiebelJI_enu.jar in the classpath as follows:

```
java -classpath <full_path_of_SiebelJI_Common.jar>;<full_path_of_SiebelJI_enu.jar>;. DataBeanDemo
```

Standalone Visual Basic Program Code Sample

This standalone Visual Basic program has been built using Siebel Business Object API. To run the Visual Basic program, follow these steps:

1. Save the following Visual Basic code in a file and name the file "check_connection.vbs".

```

-----Sample Program Start-----
Dim errCode
Set SiebelApplication = CreateObject("SiebelDataControl.SiebelDataControl.1")
'SiebelApplication.Logging
"host=" "siebel://<gatewayserver>/<enterpriseServer>/<ObjMgr>/<SiebelServer>" " ", "<username>",
"<password>"
' ##### change me - begin #####
SiebelApplication.Login "host=" "siebel://mySiebelServer1"/siebel/SCCObjMgr_enu/SiebelServer" " ",
"SADMIN", "SADMIN"
' ##### change me - end #####
errCode = SiebelApplication.GetLastError()
If errCode <> 0 Then
ErrText = SiebelApplication.GetLastErrorText
MsgBox ErrText
else
MsgBox "Connection OK"
End If
Msgbox "Done"
-----Sample Program End-----

```

2. Open the check_connection.vbs program and modify the section between the "change me - begin" and "change me - end" comments to include the connection parameters of your Siebel Server.
3. Save the file.
4. Double click to run it.
A pop-up message appears to notify you whether the connection has been established or has failed.

Adapters - Websphere MQ Adapter

Common Problems and Typical Causes

Installation and Startup Issues

WebSphere MQ Adapter Link Does Not Appear in the Adapters Panel after Installation

After completing the installation for WebSphere MQ Adapter, if the WebSphere MQ Adapter link does not appear in the **Integration Server Administrator > Adapters** panel, follow these steps:

1. Check for startup errors in the Integration Server server.log. For example, you might see the following error:

```
2009-05-10 17:39:18 CEST [ISS.0028.0012C] WmMQAdapter: Startup service
(wm.mqseries.admin:registerAdapter)
2009-05-10 17:39:18 CEST [ISU.0000.9999E] Unable to initialize resource bundle manager
(com.wm.adapter.wmmqadapter.wmMQAdapterResourceBundle).
Can't find bundle for base name com.wm.adapter.wmmqadapter.wmMQAdapterResourceBundle, locale
en_US
at com.wm.adk.WmAdapter._init(WmAdapter.java:85)
at com.wm.adk.WmAdapter.<init>(WmAdapter.java:55)
at com.wm.adapter.wmmqadapter.wmMQAdapter.<init>(wmMQAdapter.java:75)
at com.wm.adapter.wmmqadapter.wmMQAdapter.getInstance(wmMQAdapter.java:97)
at com.wm.adapter.wmmqadapter.admin.Service.registerAdapter(Service.java:78)
...
```

2. Ensure that the necessary WebSphere MQ client library files (for example, the `com.ibm.mq.*` files) have been copied over to `SoftwareAG_directory/IntegrationServer_directory/packages/WmMQAdapter/code/jars` directory. For more information, see the webMethods WebSphere MQ Adapter Installation Guide. The exact files that you must copy over will vary depending on the WebSphere MQ version (for example, 5.x or 6.x). Such errors are typically an indication that the client libraries are missing or their version is incorrect (that is, it does not match exactly the WebSphere MQ Server version).

Connection Errors

Unable to Connect to WebSphere MQ Queue Manager or Obtain Queue Names

Symptom 1: *(appears at the top of the WebSphere MQ Adapter connection configuration page when trying to connect)*

```
Unable to obtain Queue Names:
[ART.114.2055] 114.2055 [ADA.600.2062] PCFQuery could not connect to the Queue Manager VMWare_QM -
reasonCode=2009
```

Symptom 2: *(appears at the top of the WebSphere MQ Adapter connection configuration page when trying to connect)*

```
Unable to obtain Queue Names:
[ART.114.2055] 114.2055 [ADA.600.2062] PCFQuery could not connect to the Queue Manager QM_vmware1
- reasonCode=2059
```

Symptom 3: *(appears in the WebSphere MQ Adapter connection configuration Available Queues field)*

```
"The list of Available queues could not be retrieved from the Queue Manager"
```

To resolve such connection issues, go through the following checklist:

- Verify that the **Channel Name** matches the spelling and case on the WebSphere MQ Server.
- Verify that the **Channel Name** on the WebSphere MQ Server is of type **Server Connection**. (This is only necessary if you are running the WebSphere MQ Server on a different machine than WebSphere MQ Adapter.)
- Verify that a different client program can establish connection to that WebSphere MQ Queue Manager using the same settings as WebSphere MQ Adapter.
- Check the WebSphere MQ trace logs and WebSphere MQ Adapter logs to see if they contain an WebSphere MQ reason code. If you find an WebSphere MQ reason code, see the IBM MQ Programmer's Reference Guide follow the instructions there to correct the issue.
- If you are connecting to the WebSphere MQ Server as a remote server, ensure that:
 - The network IP address is correct.
 - The WebSphere MQ port is correct. If you do not specify a port, port 1414 is used by default.
- If you are connecting to the WebSphere MQ Server as a local server, check if you can connect to WebSphere MQ Server as a remote server.
- Ensure that the PCF service is running on the WebSphere MQ Server.

WebSphere MQ Adapter Can Not Find mqjbnd05 (mqjbnd05.dll) Library in java.library.path

If the **mqjbndnn.dll** file is not present in the Windows system PATH variable, an error message similar to the following displays in the error log:

```
com.ibm.adk.error.AdapterServiceException: [ART.114.2055]
Can't find library mqjbnd05 (mqjbnd05.dll) in java.library.path
...
```

For details and instructions how to correct the issue, see the “*webMethods WebSphere MQ Adapter Installation Guide*”.

Issues when Message Handler Uses Sync-Point

If you experience an issue when using WebSphere MQ Adapter with sync-point, go through the following checklist:

- Check if the issue still occurs when not using sync-point (that is, use an adapter connection with the "Non-Transactional" transaction setting).
- Make sure you perform a rollback or commit when writing information to the database. If your database updates do not perform a rollback or commit, the WebSphere MQ Adapter service may hang.
- If you use a PEEK operation, ensure that you have set the releaseConnection to **Yes**.

WebSphere MQ Adapter Hangs on Put or Get Operation

In some instances, running the WebSphere MQ Adapter using sync-point can stop the processing of a PUT or GET operation.

To resolve this issue, call a rollback or a commit operation when using sync-point.

Messages Sent to or Received from WebSphere MQ Server Appear Corrupted

If you encounter corrupted messages that are sent to, or received from, the WebSphere MQ Server, do the following:

- WebSphere MQ Adapter uses the CCSID value specified in the message handler settings. If you have not specified a value for CCSID, see the webMethods MQSeries Adapter Users Guide to determine the default CCSID the adapter uses.

Message Not Reaching Destination Queue After PUT Operation

If you encounter problems with messages not reaching their destination queues after a PUT operation, check the following:

- The dead-letter queue to see if messages are there.
- The logs to see if the PUT operation completed successfully.

Message Not Picked Up by WebSphere MQ Adapter After GET, PEEK, or LISTEN Operations

If you encounter problems with the adapter not picking up messages after a GET, PEEK, or LISTEN operation, do the following:

- If you did not specify a filter, the WebSphere MQ Adapter picks up the first message in the queue. Ensure that the queue is not empty.
- If you have specified a filter, verify that the message with the filter criteria is present on the queue. For example, if WebSphere MQ Adapter filters messages based on their correlation ID, ensure that the headers of the messages in the queue are populated with correlation ID.
- Try to execute the operation with a client other than WebSphere MQ Adapter to ensure that messages are on the queue and that they can be picked up from the queue.

Information to Provide When Opening a Support Incident

When you open a Support Incident related to WebSphere MQ Adapter, provide the following information to Software AG Global Support:

- Typical Integration Server environment information (run `wm.server.admin:getDiagnosticData` to collect logs, config files, and fix levels).
- Whether the Integration Server is part of a cluster.
- WebSphere MQ Adapter details (from the Adapter's About page or the Package Home page provide version, build, and fixes applied).
- If possible, the WebSphere MQ Adapter trace logs (see the next section for instructions).
- Screenshots of the WebSphere MQ Adapter connection, listener, or notification (if relevant to the problem).
- Directory listing of the Integration Server file system showing the WebSphere MQ client libraries (if this is an installation or initial connectivity issue).
- The MQSeries environment information (operating system and version).
- The MQSeries-specific error (for example, the reason code if one is present).
- Whether Integration Server is using the adapter to produce a message to MQSeries or to consume a message.
- Whether this type of operation or test worked at some point, but does not work now, or it has never worked.
- Pattern of incidence (100% of time, intermittent, or other).

WebSphere MQ-Level Trace

WebSphere MQ-level tracing can reveal low-level communication between WebSphere MQ Adapter, the WebSphere MQ client libraries, and the WebSphere MQ Server.

To enable the WebSphere MQ-level tracing:

1. In Integration Server Administrator, go to **WebSphere MQ Adapter > WebSphere MQ-Level Tracing**.
2. Set the **WebSphere MQ Trace Status** to **Enabled**.
3. Set the **Trace Level** from 0 to 5 (5 is the highest level).
4. Specify a **Trace Filename**.

After enabling WebSphere MQ-level tracing, reproduce the issue so that information is written to these logs.

The WebSphere MQ Adapter returns exceptions from the queue manager on the WebSphere MQ Server if an issue is encountered (for example, an exception occurs because an invalid parameter is passed to the WebSphere MQ Server).

Directory Listing of the MQ Client Jars and Props Files

Provide a directory listing of the directories where .jar and .prop files are installed.

Zip of Adapter Package

(Optional) Provide a zip file containing the WmMQSeries or WebSphereMQ packages.

Corrective Actions

- Ensure that the WebSphere MQ client libraries are the same version as that of the MQSeries Server.
- Ensure that you have installed the correct support packs for the version of WebSphere MQ Adapter you are running.
- Ensure that you installed the WebSphere MQ Adapter fixes in the correct order.

B2B products - Trading Networks

Troubleshooting Information

From Trading Networks 7.1 version onwards, you can use My webMethods Server (MWS) to create profiles and to monitor transactions, tasks, and activity logs. These functions have since been deprecated in the Trading Networks Console. However, you can enable the deprecated functions and use them in the Trading Networks 7.x and 8.x versions using the View menu in the TN Console.

Common Problems

Trading Networks Database Usage

The JDBC connection pool that Trading Networks uses is defined in the Integration Server Administrator > Settings > JDBC Pools page. To find details about how many connections are in use, or are available, and so on, you can use the `wm.server.admin:getDiagnosticData` utility service (`http://localhost:5555/invoke/wm.server.admin/getDiagnosticData`). This service creates the `<DiagnosticData.zip>/config/JDBCPools.txt` and `<DiagnosticData.zip>/runtime/JDBCPools.txt` files that provide the required details. In addition, it is also necessary for a database administrator to occasionally examine the database server.

The Trading Networks JDBC pool used the DataDirect Connect for JDBC driver like all other inherent IS JDBC components. This driver is packaged in the `<webMethods_directory>/common/lib/ext/sl53_cj36.jar` file. To find the driver details such as the version or fix level, do the following:

1. Open the `sl53_cj36.jar` file.
2. Locate a file that has the name in the following format
`"jdbcreadme_<versionnumber>_<buildnumber>.txt"`.
 For example, `'jdbcreadme_36_0007.txt'`. This file indicates that the driver version is 3.6 and the build number is 0007.

In addition, it is also necessary to look into a fix or upgrade the driver to correct an issue.

It is also sometimes useful to enable the debugging at the driver level (referred to as Spy logging in the CJDBC driver). Refer to [Database Troubleshooting](#) for details.

Viewing or Mapping Certain Trading Networks Objects in the Pipeline of FLOW Services

There are certain Trading Networks objects that webMethods Developer does not correctly parse and make them easily readable. As a result, when viewing the pipeline contents from Trading Networks services such as `wm.tn.doc:view` or `wm.tn.queueing:getRegisteredQueues`, it is not uncommon to see something as below in the Developer > Results view:

```
>>>BasicData:PartName=EDIData,MimeType=application/EDI; charset=UTF-8,Length=2182,Bytes=
B@247418,PartIndex=0,StorageType=null,StorageRef=null,LargePart?=>>>BasicData:MBoolean=false<<<<<<
```

To correct this, you must add some Trading Networks libraries in the Developer Classpath as follows:

1. Edit the `<webMethods_directory>/Developer/config/ini.cnf` file.
2. Add the `<webMethods_directory>/IntegrationServer/packages/WmTN/code/jars/static/tncore.jar` file to the classpath as below:

```
...
application.classpath=...;C:\\<webMethods_directory>\\IntegrationServer\\packages\\WmTN\\code\\
\\jars\\static
tncore.jar
...
```

3. Save the file and restart Developer.

4. In Developer > Support Information > Developer Classpath, you will see the `tncore.jar` file listed. With this, when running WmTN services, all the objects are expandable and mappable.

Trading Networks Cluster Configuration

If you are using a Trading Networks cluster, do not store queryResults in the session object. Ensure this by setting the following property in the `properties.cnf` file of each Integration Server/Trading Networks node in the cluster: `tn.query.threshold=-1`

Controlling Thread Usage for Trading Networks Tasks

The property `tn.task.threadpool.pct` controls the percentage of the Integration Server's pool of threads (that is, from `watt.server.threadPool`) that the task manager can use to process tasks simultaneously. The default value for this property is `.5` (that is, 50% of the value for `watt.server.threadPool`). Trading Networks enforces a range of `.25` to `.8` as valid values for this property. However, if you set the value to less than or greater than the range, the value is reset to `.25` or `.8`, and a warning is logged.

It has been observed that sometimes, an outgoing delivery (that is, an HTTP delivery) gets hung at some given endpoint. This occurs because of an abnormal activity such as accepting the connection but not writing a response and so on, at the remote HTTP server. Due to this, the Integration Server's HTTP client does not timeout the connection (`watt.net.timeout=0`) by default. In such a scenario, when

Trading Networks is delivering many documents to the same endpoint, and one of the endpoints gets hung, the HTTP client services continue to pile up each as a hung thread. You can see this by doing either of the following:

1. Look up the `Integration Server Administrator > Service Usage` page. Here, you may see services such as `pub.client:http(25)`, where 25 indicates that 25 instances are currently running.
2. Look at a Java Thread Dump. Here, you will see several threads simultaneously waiting for a response from a remote server.

It is recommended that you change the `watt.net.timeout` property to some finite value so that the HTTP services do not hang indefinitely. A timed out request is better than restarting Integration Server to clear the hung threads. In Integration Server 8.x versions, however, the ability to stop or kill running threads has been implemented.

Trading Networks Query Results Accumulating on the File System

From Trading Networks 7.1 version onwards, Trading Networks temporarily stores the caching files (commonly known as "queryResults") in a configurable directory on the server. All these files have a coherence* prefix. The location is determined by the `tn.tmpdir` property. If you do not specify the `tn.tmpdir` property, the value of `java.io.tmpdir` is used. In earlier versions of Trading Networks, these TN queryResults are stored in Integration Server's Repository2 file or the database location.

Note: Integration Server's Repository2 is no longer available from Integration Server 7.x version onwards. For earlier versions, it is recommended that you periodically run the `wm.tn.enumerate:deleteQueryResults` service to purge the accumulated cache files. You can schedule this service using the Integration Server scheduler.

Access Problems in My webMethods Server Trading Networks Pages

From 7.1.2 version onwards, Trading Networks supports Single Sign-On (SSO) to enable the communication between My webMethods Server and Trading Networks/Integration Server. To enable this hybrid SAML, set the following Integration Server configuration property:

```
watt.server.auth.samlResolver=http://host:8585/services/SAML (8585 is the default port of the My webMethods Server)
```

If a My webMethods Server user cannot access several Trading Networks pages (that is, `Monitoring > Integration > B2B > Transactions` or `Administration > Integration > B2B > Partner Profiles` page), you may notice the following in the Integration Server's server.log:

```
[ISS.0053.0002C] Access denied for user SAMLart on port 5555 -> 'soap/rpc' from xx.xx.xx.xx
[ISS.0012.0011W] Resolution of SAML artifact "xxx" failed
```

To troubleshoot this issue, do the following:

- Ensure that My webMethods Server is running on the host and the port specified in the Integration Server's `watt.server.auth.samlResolver` property.
- Verify that Central User Management is enabled on Integration Server. Its JDBC pool must point to the same schema shared by the My webMethods Server via its `<webMethods_directory>/MWS/server/default/config/mws.db.xml` file. If Central User Management is enabled, on Integration Server Administrator > Security > User Management page, you will see Central User Management = "Configured".
- Create a new user on My webMethods Server and assign it to the `TNAdministrators` role, and then check if the user can access the Trading Networks pages.
- Check the ACLs, Groups and Users settings in Integration Server to ensure that the new user is a member of the `TNAdministrators` group with `TNMWSUsers` ACL.
- Check if a user exists in both the local directory and the central directory (usually "Administrator") and verify if these have different passwords. This could cause issues.

java.io.FileNotFoundException While Trying to Retrieve a Trading Networks "large document"

When calling `wm.tn.doc:getDeliveryContent` or `wm.tn.doc:getContentPartData` services, occasionally you may encounter a `FileNotFoundException`, and see the following in the Integration Server's `server.log`:

```
Error retrieving content of content part ffddata of document xxx.
The exception message is : /xxx/#DocRes.dat (No such file or directory).
The stack trace is : java.io.FileNotFoundException: /xxx/2DocRes.dat (No such file or directory)
at java.io.FileInputStream.open(Native Method)
at java.io.FileInputStream.(FileInputStream.java:106)
at java.io.FileInputStream.(FileInputStream.java:66)
at com.wm.util.tspace.Reservation_FileImpl.getInputStream(Reservation_FileImpl.java:81)
```

To troubleshoot this issue, you must verify if you enabled large document handling, the values set for the relevant parameters, and so on:

- Check if you configured the following property so that Trading Networks considers certain documents as "large documents":

```
tn.BigDocThreshold=.
```

You can check this property either in the

`<webMethods_directory>/IntegrationServer/packages/WmTN/config/properties.cnf` file, or on the Integration Server Administrator > Solutions > Trading Networks > Settings > TN Properties page, or on the My webMethods Server > Administration > Integration > B2B Settings > Server Settings page.

- Check if you configured the following properties so that Integration Server handles "large documents":

```
watt.server.tspace.location=
watt.server.tspace.max=
watt.server.tspace.timeToLive=
```

You can check these properties either in the

<webMethods_directory>/IntegrationServer/server.cnf file, or on the Integration Server Administrator > Settings > Extended Settings page.

Often, you may need to extend the time-to-live interval so that the #DocRes.dat file can stay in the tspace directory for a longer duration before Integration Server purges it. The default value is 0, which means that the file will be deleted when a new reservation file is written in the tspace directory.

Issues with Data Migration Using the Import and Export Utilities

Import and export operations for Trading Networks data migration must be carried out in steps. By default, Trading Networks does not consider dependencies of references when importing data. To avoid import errors, you must import data in a specific order.

For example, if you try to export Trading Networks data all at once, you may encounter one or more of the following errors:

```
java.sql.SQLException: [wm-cjdbc33-0009][SQLServer JDBC Driver][SQLServer]INSERT statement
conflicted with COLUMN FOREIGN KEY constraint 'fk_BDocTypAtt_AttId_BDocAttDef'. The conflict
occurred in database 'webmethods2', table 'BizDocAttributeDef', column 'AttributeID'.
```

```
ERROR An error occurred while attempting to import partner profiles; error information follows.
class: "java.sql.SQLException", message: "Attempted to save an external ID that has an unknown
type (type code=106)."
```

```
ERROR An error occurred while attempting to import extended field definitions;class:
"java.lang.Exception", message: "Attempted to import extended field "XYZ" for partner ID XXXXXX,
but this partner does not exist on your Trading Networks system.", stack: "java.lang.Exception:
Attempted to import extended field "XYZ" for partner ID XXXXXX, but this partner does not exist on
your Trading Networks system. This error occurs when foreign key constraints are violated because
definitions or referred data types were unavailable on the target machine.
```

To avoid Trading Networks migration errors when using the import/export utility, do the following:

1. Export field groups, field definitions, external ID types, and partner data (profiles) for Enterprise and trading partners **in separate files**.
2. Export document types.
3. Export processing rules.
4. Import those files on the target machine, **in the order in which you exported them**.

Recommendations:

- Keep server configuration (version, patches, and fixes for Integration Server, Trading Networks, etc.) the same on source and target machines.
- If you are using the EDIINT module, disable the package on the target machine. Then, import the files and enable the module. This avoids the conflicts in Internal IDs that are automatically generated on each Integration Server by propagating the same internal IDs on the source and target machines.
- Ensure that Internal IDs are synchronized in development, testing, and production environments.

How to Grant My webMethods Server Users Access to Limited Data Sets

- In My webMethods Server > Administration > System-Wide > Roles, create a new role "SpecialTNRole" (for simplicity use Static Role Provider, but you can also use the others).
- In My webMethods Server > Administration > System-Wide > Users, create a new user "SpecialTNUser". Then, modify SpecialTNRole to have SpecialTNUser as a member.
- In My webMethods Server > Administration > System-Wide > B2B Permissions > Data Permissions > Add Data Set > DataSet tab, select the required Partners/DocTypes the user(s) must see, or select "All" for each.
- On the Permissions tab, add the SpecialTNRole role, and then grant the desired B2B permissions.
- Verify the setup by logging out of My webMethods Server and logging back in as SpecialTNUser. This user should only be allowed to perform actions according to the permissions and the DataSet configured as above.

Information to Gather When Opening a Support Incident

Please provide the following information when logging a Trading Networks related support incident with Software AG Global Support:

Environment Versions/Fix levels

- Screenshot of the IS Administrator > About page
- Screenshot of the IS Administrator > About > Packages/Updates > View page (or `http://localhost:5555/WmRoot/updates.dsp`)
- Screenshot of My webMethods > About > Trading Networks tab

Configuration Files

- Configuration files available in the `<webMethods_directory>/IntegrationServer/packages/WmTN/config` directory:
 - `Properties.cnf` (contains general TN configuration)
 - `Dblimits.cnf` (contains Trading Networks database configuration)

Log Files

- Integration Server logs (server log, error log, and so on). These can be best captured using the `wm.server.admin:getDiagnosticData` utility service (`http://localhost:5555/invoke/wm.server.admin/getDiagnosticData`) that zips/bundles the Integration Server About page, configuration files, log files, and so on. It may also help if you step-up the Trading Networks-specific logging capabilities (that is, `{{Integration Server Administrator > Settings > Logging > Trading Networks > 0000 Database = TRACE}}`) and recapture the Integration Server server log output.
- For a TN Console issue, generate a debug log at higher debug levels using the following steps:
 - Open the `<webMethods_directory>/TNConsole/config/ini.cnf` file.
 - Change the debug level parameter from its default value (4) to a higher value. For example:

```
application.mainMethodParams=-debug 10
```

- Save the file and restart the Console at the command line redirecting the output to a file:

```
webMethods_directory/TNConsole/bin/console.bat > console_debug-10.log
```


B2B products - C1 Onramp Adapter

Introduction

The webMethods onRamp for Commerce One MarketSite Adapter is a package you install on webMethods Integration Server. It enables sending and receiving of xCBL business documents to and from Commerce One MarketSite applications. Commerce One MarketSites are portals used by businesses to exchange goods and services worldwide.

Common Symptoms

- The service `pub.marketconnect.transport:sendEnvelope` fails with external Partner.
- Received envelopes show up as unknown messages in Trading Networks and do not invoke the handler service.
- The C1 OnRamp Adapter hangs on many instances of `pub.marketconnect.transport:sendEnvelope`.
- Submitting an xCBL envelope to Trading Networks throws a database error.
- Unable to access the C1 OnRamp Adapter error logs from the Adapter home page.

Possible Causes and Resolutions

`pub.marketconnect.transport:sendEnvelope` fails with external Partner

The service `pub.marketconnect.transport:sendEnvelope` can fail due to one of the following common configuration errors:

- The used MarketSite alias configuration is not configured correctly, or the MarketSite itself is not reachable for some reason.
- If you are using HTTPS communication with certificates, the certificates configuration may be incorrect, or (one of) the certificates is/are expired.

Test the MarketSite alias to make sure the connection is successful. Verify that the certificates configuration is valid when you are using HTTPS with certificates.

Received envelopes show up as unknown messages in Trading Networks and do not invoke the handler service

This can happen if the `WmMarketConnect` package is enabled on a Production environment. Because the example envelope handler services can interfere with the custom handler services, the custom handler services may not work as expected. When you are ready to begin running the C1 OnRamp Adapter in a production environment, disable the `WmMarketConnectExample` package and reload the `WmMarketConnect` package or restart the Integration Server. This ensures that the example envelope handler services are not registered and the custom handler services are used.

C1 OnRamp Adapter hangs on many instances of `pub.marketconnect.transport:sendEnvelope`

The C1 OnRamp Adapter can get into a hanging state when running many instances of `pub.marketconnect.transport:sendEnvelope` at the same time. This is typically observed when around 100 or more instances are running. To resolve this, set the Envelope Logging level to 0. This will increase the performance of the adapter, so it doesn't hang. The consequence of this setting is that no logs are written.

- Navigate to the C1 OnRamp Adapter configuration page.
- Set the number of logged envelopes (sent/received) to 0.
- Restart the Integration Server.

Submitting an xCBL envelope to Trading Networks throws a database error

The error "[101.518] ** Exception: Unable to connect to database!*" may be seen when an xCBL document is submitted to Trading Networks 7.x or higher on an Integration Server where the WmPRT package is disabled or the Process Audit JDBC pool is not configured. To resolve this issue, apply the latest Service Pack for the Adapter and C1O_3-0_Fix39. This fix enables the C1 OnRamp Adapter to work without configuration of the Process Engine.

Unable to access the C1 OnRamp Adapter error logs from the Adapter home page

If there is a huge number of entries in the C1 OnRamp Adapter's error log, loading the errors can take a very long time and the browser might appear to be hanging. To get around this situation, you should delete all entries from the error log. The standard option to delete all entries from the error log is to use the link "Delete All" from the C1 onRamp Adapter's errors page.

When the errors page itself is no longer accessible from the browser, all entries can be deleted by entering the URL `http://<IS Host>:<Port>/WmMarketConnect/error/showErrors.dsp?action=deleteAll`

Information to Gather When Opening a Support Incident

To troubleshoot onRamp for Commerce One MarketSite Adapter related issues, obtain the following information:

- Provide the information listed on the listed on the [Integration Server Troubleshooting](#) page
- Version and build number of the C1 OnRamp Adapter, including the fix level
- Fix level for Trading Networks when using the C1 OnRamp Adapter to work with Trading Networks
- Integration Server server and error logs
- C1 OnRamp Adapter logs

Note: The C1 OnRamp Adapter logging level can be changed from the Adapter home page. Detailed information is available in the Adapter's User's Guide.

B2B products - EDI Module

Common Problems / Typical Causes

- I need to install an EDI document type, but cannot find the "Install TN Document Types" option in the EDI Module home page.
 - Make sure you have installed and enabled the "WmEDIforTN" package.
- I need to install an EDI document type, but the required EDI standard and version is not listed in the "Install TN Document Types" page.
 - Only the EDI standards and versions supported by EDI Module will be listed in the "Install TN Document Types" page. Before opening a Support Incident, please check the fixes available for supporting additional EDI standards and versions. Also, check the Knowledge Center on [Empower Product Support Web site](#) for support requests related to the document type you want to install. For more information about how to install the EDI document types, see the webMethods EDI Module Installation and User's Guide.
- I cannot open the EDI document type in Trading Networks Console. Trading Networks Console displays "No Document type editor found!" error message.
 - EDI Module is designed in such a way that you cannot edit the EDI documents using Trading Networks Console. You can edit the EDI documents using the EDI Module built-in services in webMethods Developer. For more information about the built-in services, see the webMethods EDI Module Built-In Services Reference.
- I have defined a TPA (Trading Partner Agreement) for EDI Module, but EDI Module uses the default EDITPA.
 - Ensure that you set the Agreement ID in the TPA to "EDITPA". Otherwise, EDI Module will not use the TPA defined by you.

General Recommendations

Check for the latest available fixes and install them if possible. This will reduce the possibility of encountering issues that are already corrected.

Information to Gather When Opening a Support Incident

Environment Information

List of Installed Fixes

- When you submit a Support Incident related to EDI Module, please provide the following information to Global Support:
 - A screenshot of the Integration Server's "About" page
 - A screenshot of the Integration Server's "Packages and "Updates page (Click the "View" link at the bottom of the "About" page)
 - Screenshots of the home pages for the following EDI related packages:
 - WmEDI (EDI Module Core Component)
 - WmEDIForTN (EDI Module Trading Networks)

Server & Error Logs

- In order to retrieve good logging results, you need to provide the following:
 - Server log (set to at least "Info" for Integration Server)
 - Error log
 - Developer and/or My webMethods Server Errors (if any)

Simplified Reproduction Package

For optimal assessment of the issue at hand, provide a simplified package. This package should contain the required services which clearly replicate the issue you face. You also need to provide the required inputs such as the variables and the EDI messages used.

B2B products - EDIINT Module

Common Symptoms

Any of the following symptoms may indicate that there could be an EDIINT related issue:

- Inbound EDIINT document comes into Trading Networks with unknown sender and receiver
- Not able to receive AS/2 messages when using Reverse Invoke/Reverse Gateway
- Trading Networks activity log for EDIINT transaction shows Security Check failed / Payload not processed messages
- Payload is not submitted to Trading Networks
- Payload is submitted to Trading Networks but it is not recognized by Trading Networks
- Payload is submitted to Trading Networks but its sender/receiver is unknown
- Decryption failed message in the activity log of EDIINT transaction in Trading Networks

Possible Causes and Resolutions

Inbound EDIINT document comes into Trading Networks with unknown sender and receiver

EDIINT messages come with "AS2-From", "AS2-To" EDIINT message headers. The "EDI ID Match" option affects the way how Trading Networks will recognize the Sender and Receiver of the inbound EDIINT messages. You can set "wm.EDIINT.EDIINTIDMatch" to "true" in the properties.cnf file in the <webMethods_install_directory>/IntegrationServer/packages/WmEDIINT/config folder or you can set it from home page the of WmEDIINT package by selecting the "EDIINT ID Match" option.

When "wm.EDIINT.EDIINTIDMatch" is set to "true", Trading Networks will search the sender profile which has the "EDIINT AS2" external identifier defined and it should match with the value of "AS2-From" from the EDIINT message. Similarly for the receiver profile Trading Networks will search for the profile with the "EDIINT AS2" external identifier defined, which matches the value of "AS2-To". If Trading Networks is not able to find such a profile, it will assign "Unknown" to either Sender, Receiver or both.

If "wm.EDIINT.EDIINTIDMatch" is set to "true" then make sure that profiles participating in EDIINT transactions have the "EDIINT AS2" external identifier defined. Otherwise you can set "wm.EDIINT.EDIINTIDMatch" to "false", so that Trading Networks can search the matching profile with other external identifies.

Not able to receive AS/2 messages when using Reverse Invoke/Reverse Gateway

In a Reverse Gateway setup, the Reverse Gateway server receives the request from external clients and then forwards them to the internal Integration Server for processing. In case the Reverse Gateway server is not able to forward the data that it received from a client in its original format, it can cause a problem during the recognition phase and the WmEDIINT module will not be able to recognize the EDIINT message properly. If you face this situation then please make sure that the EDIINT/EDIINT MDN payload that the Reverse Gateway server has received is identical to the one that has been received by the internal Integration Server. The problem could be in the EDIINT payload that is sent by the client and received by the Reverse Gateway server.

Trading Networks activity log for EDIINT transaction shows Security Check failed / Payload not processed messages

The WmEDIINT module only allows more secure inbound EDIINT messages. When it identifies that an inbound EDIINT message is less secure than the configuration it will not process the payload and will add a "Security Check failed" message in the activity log.

How the WmEDIINT module figures out whether an inbound EDIINT message is more secure or less secure:

1. Determine the S/MIME Type of the inbound EDIINT message, that is whether the inbound document is "plain", "signed", "encrypted", or "signed and encrypted".
2. Check the value of the S/MIME Type extended profile field of the SENDER's profile in Trading Networks.
3. Compare the S/MIME Type for the inbound EDIINT message with S/MIME Type of SENDER's profile
 - a. If S/MIME Type of SENDER's profile is set to "plain" then inbound documents can be plain, signed, encrypted, or signed and encrypted.
 - b. If S/MIME Type of SENDER's profile is set to "signed" then inbound documents can be signed, encrypted, or signed and encrypted. "plain" is not allowed because it is less secure.
 - c. If S/MIME Type of SENDER's profile is set to "encrypted" then inbound documents can be encrypted, or signed and encrypted. "plain" or "signed" are not allowed because they are less secure.
 - d. If S/MIME Type of SENDER's profile is set to "signed and encrypted" then inbound documents can only be signed and encrypted. "plain", "signed" or "encrypted" are not allowed because they are less secure.

Payload is not submitted to Trading Networks

As installed, the EDIINT Module is configured to submit the payload to Trading Networks for business level processing of the payload. When the EDIINT Module is configured to send the payload to Trading Networks, the EDIINT Module submits the payload after it completes transport-level processing.

The payload will not be submitted to Trading Networks when

- The WmEDIINT module is not configured to submit the payload to Trading networks. Please verify the WmEDIINT module configuration either from the WmEDIINT home page or in the WmEDIINT Module Configuration Properties section. Check the property "wm.EDIINT.submitPayload".
- When there is any problem encountered while the EDIINT module is doing transport-level processing. Please check the activity log entries for the specific EDIINT transaction and take corrective action to fix the transport-level processing.

Payload is submitted to Trading Networks but it is not recognized by Trading Networks

When the content-type of the payload is "application/edi-X12", "application/EDIFACT", "application/edi-consent" (for EDI documents) or "application/XML" (for XML documents), Trading Networks will be able to recognize the payload and do further business level processing. When the content-type is other than "application/edi-X12", "application/EDIFACT", "application/edi-consent" or "application/XML" you need to specify the "User Process Payload Service" that will process payloads and submit them to Trading Networks.

If you face the situation where the payload is submitted to Trading networks but is not recognized by TN, there can be a problem with the payload (EDI / XML), or your custom payload processing service is not working correctly (for custom payloads).

For EDI and XML payloads, you can try submitting the payload directly to Trading Networks to check if it's a valid payload and take the corrective actions. For custom payloads you may need to debug the custom payload processing service to make sure that it is able to handle the custom payload correctly.

Payload is submitted to Trading Networks but its sender/receiver is unknown

Please check the section "Payload is submitted to Trading Networks but it is not recognized by Trading Networks" and make sure that the payload is recognized by Trading Networks. For EDI and XML documents, Trading Networks will extract the "SenderID" and "ReceiverID". For custom document types, the gateway service is responsible for providing the guideline, so that Trading Networks can associate the senderID and receiveID with the transaction.

Decryption failed message in the activity log of EDIINT transaction in Trading Networks

While processing an encrypted EDIINT message, the WmEDIINT module might report a decryption error. Please check the following:

- When the WmEDIINT module receives an encrypted EDIINT message, it will use the private key from the receiver's profile. The private key and the public certificate that was used to encrypt the EDIINT message should belong to the same key chain, otherwise the message will not be decrypted. In order to make sure that the private key and public certificate are from the same key chain, you can use openssl to check their modulus and compare.

- Check the modulus of the private key

```
openssl pkcs8 -in privKey.der -inform DER --nocrypt | openssl rsa --modulus --noout
```

- Check the modulus of the server certificate

```
openssl x509 -in cert.der -inform DER --modulus -noout
```

- Please make sure that the JVM used by the Integration Server has JCE Unlimited Strength Jurisdiction Policy Files. You can download these files from the JVM vendor's web site and put them in the <jvm>/jre/lib/security directory of the JVM that you use to run the Integration Server. Install the files using the instructions in the vendor documentation.

General Recommendations

General Recommendations regarding the Integration Server are described in the [General Recommendations](#) section.

Information to Gather When Opening a Service Request

To troubleshoot issues related to the webMethods EDIINT module, obtain the following information:

- Provide the information listed on the [Integration Server General Information](#) page.
- The fix level for the WmEDIINT module.
- The fix level for Trading Networks.

B2B products - RosettaNet Module

Common Symptoms

Any of the following symptoms may indicate a problem related to RosettaNet Module:

- Data failed within Trading Networks with UNKNOWN DOCUMENT error
- Failure to verify signature/decrypt message
- Error in RosettaNet process model
- Outbound data cannot be parsed by a trading partner

Possible Causes and Resolutions

Data Failed within Trading Networks with UNKNOWN DOCUMENT Error

- Ensure the proper RosettaNet document type is installed in Trading Networks.
- Ensure the RosettaNet document follows the standard. Basic guidelines follow.

A RosettaNet Business Message always contains the following elements:

- A Preamble Header
- A Delivery Header
- A Service Header
- Service Content

Service Content consists of an action message or a signal message.

If Service Content is an action message, one or more attachments may be included.

Message-ID

A RosettaNet object (RNO) typically starts with two dashes followed by the value of the Message-ID element. In the following example, the value of Message-ID is "RootBoundary:"

```
--RootBoundary
Message-ID: (((<RootBoundary>)))
Mime-Version: 1.0
Content-Type: multipart/related; type="application/xml";
boundary="--RequestMessageBoundary"
Content-Description: Unsigned Message
```

An RNO typically ends with two dashes, followed by the value of the Message-ID element, followed by two more dashes. For example:

```
--RootBoundary--
```

The value specified in the **boundary** parameter within the MIME Header Content-Type is used to divide the Preamble header, the Delivery Header, the Service Header, and the Service Content.

The standard is two dashes followed by the boundary value, as shown in the following example. (Notice that in this example the boundary value is "RequestMessageBoundary" from the example above.)

```
-----RequestMessageBoundary
Content-Type: Application/XML
Content-Transfer-Encoding: quoted-printable
Content-Location: RN-Preamble
Content-ID: (((<21200562116523836_Preamble>)))
-----RequestMessageBoundary
```

The end tag for RequestMessageBoundary is two dashes, followed by the value of the boundary parameter, followed by two more dashes. For example:

```
-----RequestMessageBoundary--
```

For more information about the RosettaNet Standard, see the RosettaNet Implementation Framework Section at <http://www.rosettanet.org>.

Failure to Verify Signature/Decrypt Message

If a signed and encrypted RNO fails because of signature verification or decryption errors, the reason is usually because certificates used by the partner (to sign and encrypt the message) are not from the same set as configured in the Trading Networks partner profile. Ensure that certificates used by your partner and ones that are configured in your Trading Networks are valid and from the same key-chain.

Outbound Data Cannot be Parsed by a Trading Partner

This type of error occurs when the outbound data is valid within the webMethods system, but it cannot be parsed by the trading partner application.

To diagnose this issue:

1. Obtain the error message generated on the trading partner side. Also ask for the expected data that should work within the trading partner application.
2. Compare the expected data with the webMethods outbound data to determine the difference.
3. Check the RosettaNet standard to see if the webMethods data complies with the RosettaNet standard. If the webMethods data complies, then the issue is on the partner side and it would be the trading partner's responsibility to fix the problem. If the webMethods data does not comply, report the issue to Software AG Global Support.

Error in RosettaNet Process Model

Other common problems seen as RosettaNet issues include when a RosettaNet process model fails with an error. When a problem like this occurs, use My webMethods Server > Monitor to locate the failed step and error message. If the error points to something from the service, troubleshoot the service that was invoked by the failed process step. Sometimes the failure is caused by invalid input data. If that is the case, trace back to determine how the input data was generated.

For details on how to troubleshoot problems related to the Model or the Process Engine see the [Process Engine Troubleshooting](#) page.

General Recommendations

General Recommendations regarding the Integration Server are described in the [General Recommendations](#) section.

Install the RosettaNet Sample

Generally, to reproduce customer issues it is helpful to have the RosettaNet Sample set up on your system. The sample contains preconfigured records and services, and will execute a RosettaNet Partner Interface Process (PIP) 3A4 Manage Purchase Order on one system (both partners are simulated on the same system). You may find that running the sample is useful to help diagnose issues such as monitoring of RosettaNet processes.

For information about and instructions for running the PIP 3A4 sample that is included with the webMethods RosettaNet Module, see the webMethods RosettaNet Module Sample Guide.

Information to Gather When Opening a Support Incident

In certain cases, it is necessary to reproduce the problem in house. Please provide following information to Software AG Global support:

- RNO data (RNO data can be found in the WmRNIF11TRP\pub\archive and WmRNIF20TRP\pub\archive directories)
- TN partner profiles
- RN TPA
- Exported Customer Modeler and related packages

External Links

- RosettaNet.org

B2B products - SWIFT Module

General Information

With the introduction of version 7.1 for SWIFT, we no longer have the possibility of installing either SWIFT FIN or SWIFTNet Components separately. These have been combined into one installable product which is available in the Software AG Installer as

- eStandards -> webMethods SWIFT Module 7.1 SP1

Note: the SWIFT Module requires the eStandards -> Common Files 7.1 -> Program Files to be installed. This will result in the installation of the crucial WmEstdCommonLib 7.1 package. You need to update this package to the latest fix level after installation, otherwise you will face functionality problems with SWIFT.

Packages related to the webMethods SWIFT Module

- WmFin
This package contains the services used to implement and support the SWIFT FIN compliant functionality.
- WmFinMessages
This package is created after the import of the first FIN Message Type.
- WmSwiftCommon
This package contains common services that are used by different SWIFT packages.
- WmSwiftNetClient
This package contains the elements (flow services, Java services, record descriptions, and wrapper services) that support webMethods SWIFTNet client-side functionality.
- WmSwiftNetServer
This package contains the elements (flow services, Java services, record descriptions, and wrapper services) that support webMethods SWIFTNet server-side functionality.
- WmEstdCommonLib
This package contains generic services that enable you to use various eStandards Modules with webMethods Integration Server.

For further information about the packages, services and how SWIFT works, please refer to the webMethods SWIFT Module Installation and User's Guide.

Common Problems and Resolutions

SWIFT FIN Message Version Not Available

Software AG releases the latest SWIFT MTs a few months before the official release date from SWIFT. This provides customers the time to test and work with the new versions before moving them to production.

New versions are usually delivered with SWIFT service packs or fixes. Review existing SWIFT service packs and fixes before opening a support incident regarding missing versions. You can refer to Empower for the latest SWIFT fixes or the Software AG webMethods Communities for announcements:

<http://communities.softwareag.com/ecosystem/communities/public/webmethods/>.

My Integration Server version is 8, but the SWIFT packages are version 7.1

The Integration Server version number does not imply that all installed packages should also have the same version number. In the case of the SWIFT packages you will see that most of them are version 7.1. Due to this, you may have to install fixes for version 7.1. For example "ESTDCOMMONLIB_7.1_Fix10".

Loading the SWIFT packages results in an error

This might be due to several reasons but the most straight forward thing to initially check is if you have the latest EstdCommonLib fix installed.

This fix resolves several defects, therefore it is important to have it installed from the start. Also check the available SWIFT fixes for any related description in the readme file.

If the latest fix does not resolve the issue, then contact Software AG Global Support for further assistance.

Unable to import new SWIFT Messages

Running the SWIFT Module 7.1 service "wm.fin.dev:importFINItems" fails with the exception

```
com.wm.app.b2b.server.ServiceException: importFINItems: Exception -
com.wm.app.b2b.services.DocumentToRecordService.bind (Lcom/wm/lang/xml/Node;)Ljava/lang/Object;
```

The issue should be resolved by installing the latest EstdCommonLib fix.

Generated Document type or Network Validation Rule is incorrect

When you import a new FIN item a document is generated in the WmFINMessages package, along with other items such as Network Validation Rules, if they are present for this MT type. If this newly generated document type or Network Validation Rule does not match the SWIFT standard description, it causes a problem later on when SWIFT services are triggered.

If this happens, check if the latest SWIFT fix provides a resolution for it. If not, then contact Software AG Global Support for further assistance.

Unable to Parse the FIN Message

The Parse Error is a common error when working with SWIFT FIN messages. The problem could be either related to the webMethods-provided MT definitions/schema or due to the FIN message which is being parsed.

- If the error resulted from the service `wm.fin.format:convertFINToIData`, make sure the "Relaxed" flag is set to "TRUE."
Setting this flag to "TRUE" allows the parsing process to go more smoothly and to overlook minor errors.
- If the error is related to subfield parsing, perform the following steps to extract the subfield:
 1. Import the FIN message and set the "subfield" flag to "TRUE";
 2. Convert the FIN message to IData;
 3. Extract the subfield from the FIN IData using an appropriate format conversion service, such as `wm.fin.dfd:convertTagFormat`.
For more information about these services, refer to the webMethods SWIFT Module Installation and User's Guide.
- Verify the validity of the FIN message by confirming it matches the standard set by SWIFT.
If you find the message to be valid, contact Software AG Global Support. Provide a reproduction sample/scenario and a copy of the SWIFT standard document on which it is based.

Unable to import the MT into Trading Networks

The webMethods SWIFT Module can work with or without Trading Networks. If you use Trading Networks, it is required to create the correct document types, processing rules and trading partner agreements (TPA). This is done automatically when the parameters "createDocType", "createProcessingRule" and "createTPA" of the service `wm.fin.dev:importFINItems` are set to "TRUE".

If you still face a problem with the Trading Networks import after setting these parameters, contact Software AG Global Support for further assistance.

The meaning of "Error Code : .."

SWIFTNet Server/Client can return errors with a certain code. For example

```
com.wm.app.b2b.server.ServiceException: Error returned from WmSwArguments; Error Code: 12
```

To understand what this Error code means, refer to the "SWIFTNet Server and Client Errors" section located in the webMethods SWIFT Module Installation and User's Guide, Appendix A. This section also describes other number codes that might come up.

Am I required to have the webMethods WebsphereMQ Adapter?

The webMethods WebsphereMQ Adapter is required if you want to use the MQHA Transport for SWIFT FIN MT Messages.

Integration Server startup is blocked by the webMethods SWIFT Module

The webMethods SWIFT Module 7.1 without Service Pack 1 can block the Integration Server startup sequence. This can happen because the WmSWIFTNet server module will try to bind the server instance to the RMI registry, and tries to look up this server instance in the registry after binding. If there is a problem during the bind the WmSWIFTNet server module keeps looking for the server instance indefinitely, which blocks the Integration Server startup sequence.

The Integration Server server will contain the error message

```
[ISU.0000.9999C] Remote Process Information for WmSWIFTNetServer
```

With webMethods SWIFT Module 7.1 SP1, the WmSWIFTNet client or server modules will not hinder the Integration Server startup. Two parameters have been introduced for this: retry count and retry interval.

If there is an exception during the bind, the WmSWIFTNet server module stops looking up the server instance after the numbers of attempts specified by the retry count. The retry interval is the amount of time WmSWIFTNET server module sleeps before looking up the server instance again. The exception will be logged in the SWIFTNet server instance logs and the Integration Server will continue starting up.

The retry count and retry interval parameters are stored in the following files:

- <SWIFT client>/config/rmiconfig.properties
- <SWIFT server>/config/rmiconfig.properties

Logging output for SWIFT Server and Client is hard to understand

In order to resolve this the logging has been enhanced with webMethods SWIFT Module 7.1 SP1.

The "SWIFTNet Server and Client Errors" section located in the webMethods SWIFT Module Installation and User's Guide, Appendix A, contains a list of error codes that will help understanding the logging output.

When contacting Software AG Global Support, provide your Integration Server server log and error log, and the process instance logs from the webMethods SWIFT Module. Individual server instance logs for the process instances are stored in the following location

- <IntegrationServer>\packages\WmSWIFTNetServer\pub\logs

General Recommendations

- Make sure the webMethods SWIFT Module is correctly configured to connect SWIFT to all related components by following the guidelines from the SWIFT Module Installation and User's Guide.
- Install the latest eStandards Common Libraries fix (for example, "ESTDCOMMONLIB_7.1_Fix10").
- Review available SWIFT Module fixes and service packs to obtain new releases of SWIFT documents or to see whether a fix was created to correct the error you have encountered. Note that fixes are cumulative and it is recommended to install the latest available fix for the webMethods SWIFT Module.
- Refer to the webMethods SWIFT Module Installation and User's Guide for further information and full explanation how the product works.

Information to Gather When Opening a Support Incident

When you submit a Support Incident related to a SWIFT issue, please provide the following information to Software AG Global Support.

Environment Information

List of Installed Fixes

- Screen shot of the Integration Server About page
- Screen shot of the Integration Server About > Packages/Updates page

Integration Server Server and Error Logs

- Server log (set to at least "Info" for Integration Server)
- Error log
- Developer and/or My webMethods Server Errors (if any)

Simplified Reproduction Package

- A simplified package that contains the required services that clearly replicate the problem you experienced.
- The required inputs (variables and MT FIN messages) used by the services in your package.

BAM Products - Optimize

Diagnostics

Logs Locations

- <webMethods_install_directory>\Optimize\Analysis\logs
- <webMethods_install_directory>\Optimize\DataCollector\logs
- <webMethods_install_directory>\Optimize\Prediction\logs
- <webMethods_install_directory>\InfrastructureDC\logs

How To Change The Logging Level

From The User Interface In My webMethods Server

Modify the Environment definition and deploy changes. For example in version 7.x, go to:
"System Wide > Environments > Define Environments > <current environment> > Configure Servers > Analytic Engine > Journal Logging".

You will need to uncheck the checkbox if you want to change the settings for Analytic Engine only.

Health Checks For Optimize

- Check if the Environment is defined and deployed.
- Check if the Analytic Engine is running.
- Check if the Analytic Engine is reachable from the user interface in My webMethods Server.

Health Checks For Optimize for Infrastructure:

- Check if the Infrastructure Data Collector is running.
- Check if the Infrastructure Data Collector is configured to connect to the same Broker as the Analytic Engine.
- Check on monitored Integration Servers if the packages "WmISExtDC" and "WmISArtDC" are enabled.

How To Check If The Database Is Correctly Installed And Reachable

- In the Database Component Configurator, run the catalog command on the database that is defined in the DB pool used by the Analytic Engine.

Common Problems / Typical Causes

Massive Data Purge Is Not Working

- Check the operation log ("operation_log") for information about the failure.
- Before troubleshooting Massive Data Purge issues, please make sure you have installed the latest DCC fix (e.g. DCC_7-1-2_Fix2).

If The Operation Log Does Not Give The Reason Of The Failure

You need to clear the existing operation logs, install the updated script that you obtain from Software AG Global Support and run Massive Data Purge again.

Note that this script will not resolve the problem, but it prints out the exact failure message to the operation log. This will help Software AG Global Support solve the problem. Also the failed jobs are notified to the main procedure so that it doesn't need to wait a long time for failed jobs.

The script can be installed from a SQL command prompt (for example: SQL> @c:\fullinsertjobmgmt.pkb).

Once Massive Data Purge is completed, export the result from the following query:

```
select * from operation_log order by ctrl_insert_dt
```

Insufficient User Privileges

Check with your database administrator what privileges are assigned to the database user that is executing Massive Data Purge. Make sure these privileges are assigned directly to the user and not through a role. Adding "Grant Create Table" and "Grant Create Trigger" will resolve the issue.

Cannot See My KPI

After creating a Key Performance Indicator (KPI), sometimes you cannot see collected data for it in the My webMethods Server user interface. Possible reasons for this include:

- Data is not pushed through
 - This can happen if facts are not collected.
- KPI is not displayed after creation
 - This can happen if you specified a long collection interval. It will then take a long time for the KPI to be displayed.
- There is a lot of data in Analytic Engine
 - If Massive Data Purge was never run, KPI calculation takes a long time.
- The database is corrupted
 - If Massive Data Purge is aborted, the database can be left in an inconsistent state.

Cannot Connect To Analytic Engine

- Check where the Analytic Engine is defined by navigating to the "Environment" definition in the My webMethods Server user interface.
- Observe the hostname and port number on the "Map Endpoints" tab. Verify that this is the same as on the My webMethods Server Settings page.
- Check with "netstat" if there is an application listening on the Analytic Engine's port. Check with "telnet" if you can connect to this port from the host where you are running My webMethods Server.
- Check if the Analytic Engine is running by pointing a webbrowser to `http://<AE host>:<AE port>/services/monitor.wsdl`. The browser should display the WSDL file.

If the Analytic Engine fails to start, you can try the following as a final resort.

- Edit the file "`<webMethods_install_directory>\Optimize\Analysis\conf\system\EndpointRegistry.xml`" to remove all the endpoint hosts, except the "Configuration Agent".

```
<?xml version="1.0" encoding="UTF-8"?>
<endpoint-registry>
<endpoint host="correct.host.here.com" name="Configuration Agent" port="15000"
protocol="https" />
</endpoint-registry>
```

- Start the Analytic Engine.
- Redeploy the Environment from the My webMethods Server user interface.

Rules Not Firing

- Check if the facts are collected for the KPI on which the Rule depends.
- Install latest Optimize Fix.
- Change rule description and save it. See if it helps.

Common Problems / Typical Causes in Optimize For Infrastructure

Asset Is Not Discovered

See if the asset is running and reachable from Infrastructure Data Collector. Check in the Infrastructure Data Collector logs if there are any errors.

Cannot See Some Monitors (KPI Instances)

Rediscover the asset.

Old Monitors (KPI Instances) Visible In Analytics Overview

Monitor names are in the column "STRING_ID" in the "DG_ANALYSIS_MONITOR" table.

- Locate the monitors you want to keep visible
- Save them to a text file (one entry per line)
- Stop the Analytic Engine and run the cleanup script
`"<webMethods_install_directory>\Optimize\Analysis\bin\cleanupMonitors.bat <monitor file name>"`

Rule Violations Are Displayed For Components That Are No Longer Monitored

For example: We stopped monitoring the Intergration Server at 10.20.111.6:5555. We have a lot of violations like:

```
Integration Server Service Failure Status
Host.HostName.10.20.111.6.IntegrationServer.Port.5555.IsPackage.Name.WmPublic.IsService.Svc.pub.sm
ime:createSignedData
```

To remove these violations, you first need to find out the Rule Dimensions by Rule Name from the "DG_WAREHOUSE_RULE_DIM" table. Use the columns "DG_WAREHOUSE_RULE_DIM_ID" and "RULE_NAME".

Example:

```
select DG_WAREHOUSE_RULE_DIM_ID from DG_WAREHOUSE_RULE_DIM where RULE_NAME LIKE
'%HostName.10.20.111.6%';
```

With these id's you can update the table "BAM_RULE_VIOLATION". Use the column "OPEN_VIOLATION".

Example:

```
update BAM_RULE_VIOLATION set OPEN_VIOLATION = 0
where DG_WAREHOUSE_RULE_DIM_ID in (select DG_WAREHOUSE_RULE_DIM_ID from DG_WAREHOUSE_RULE_DIM
where RULE_NAME LIKE '%HostName.10.20.111.6%')
```

Information To Gather When Opening A Support Incident

If you have to open a Support Incident related to an Optimize issue, please provide the following information to Software AG Global Support.

Environment Details About Where the Problem Is Occurring

- Versions of webMethods components involved
- List of installed fixes
- Operating system information for each component
- Host names and port numbers for Analytic Engine, My webMethods Server, Integration Server, and monitored machines
- Log files from the affected Optimize components.

Configuration Files

Configuration files are located in <webMethods_install_directory>\Optimize\Analysis\conf.

The valid configuration should be defined in the "Environment" section of the My webMethods Server user interface. The recommended solution for configuration problems is rather to redeploy the configuration than to change the files.

BPM Products - CAF

Introduction

In the Software AG Designer development environment, the Composite Application Framework (CAF) is used to build portal pages and web applications that are hosted in the My webMethods Server. The pages and applications rely heavily on the use of Javascript for their visual and user interface behavior. Connections to the backend are made via web services, connecting from the My webMethods Server to the Integration Server.

Common Problems / Typical Causes

The CAF controls such as Search Bar and PrincipalPickerDialog have a memory leak.

For each request to a My webMethods Server page containing these controls, a new Java bean is created with scope set to session. This causes the memory leak that results because every request generates a Java bean that occupies memory as long as the session is valid. In order to solve this problem, you must install a recent MWS fix to update the common libraries.

Strange behavior when using Internet Explorer version 6

Many visualization and performance issues occur when using Internet Explorer version 6. These are caused by the immature Javascript implementation of this Internet Explorer browser. The Javascript implementation will not be improved, because Microsoft does not support Internet Explorer version 6 anymore.

If you have no alternative and must use Internet Explorer version 6, verify that you have at least Internet Explorer 6 Service Pack 2, and install the latest Designer and My webMethods Server fixes.

General recommendations

- Review the latest available fixes and install the ones that apply.
- Check for errors in the logs
 - For more information look at the section “*Log File Associated with the Problem Component*” below.

Information to Gather When Opening a Support Incident

Provide the following information when submitting a CAF-related Support Incident to Software AG Global Support.

- Environment Details about where the problem is occurring
 - List of installed fixes for My webMethods Server and Designer
 - The CAF project exported from Designer
 - The Integration Server package or packages containing the services needed for the CAF project to run
 - Diagnostic information
 - All applicable My webMethods Server, Designer or Integration Server logs
- The specific logs required depend on where you can see the error.
- To capture the error you might need to restart Designer in debug mode or increase the log levels in My webMethods Server and Integration Server.

Environment Details about Where the Problem is Occurring

- Name and version of the browser in which you can see the issue occurring.
 - In order to get this information you can Click Help > About.
- Name and version of the application server in which the CAF project is deployed.
 - For example: My webMethods Server or Tomcat,

List of Installed fixes

- [My webMethods Server fixes](#)
 - [Designer fixes](#)
-

Use My webMethods Server Development Tools to get diagnostic/performance information.

Some Web Applications that can get diagnostic or performance information are located in "`<webMethods_install_directory>\MWS\components\development\tools`". The SOAP monitor is very useful; using this Web Application you can see all the SOAP requests and responses that are sent and received by the My webMethods Server.

In order to use the SOAP Monitor you need to copy "`<webMethods_install_directory>\MWS\components\development\tools\wm_soapmonitor.pdp`" to "`<webMethods_install_directory>\MWS\server\default\deploy`". Next, deploy the Web Application to My webMethods Server.

- If auto-deploy is on, the SOAP Monitor is automatically installed.
- If auto-deploy is off, log into My webMethods Server as sysadmin and use "Install Administration" to deploy the new pdp.

Access the SOAP Monitor by pointing a browser to `http://<MWS_host>:<MWS_port>/portlet/wm_soapmonitor`".

Follow the same procedure to install the other Web Applications.

Log File Associated with the Problem Component

- [MyWebMethods Server](#)
- [Designer](#)
- [Integration Server](#)

For troubleshooting CAF issues, you might need to increase the Integration Server logging levels for the SOAP facility. You can add the following extended settings:

- `watt.server.WSDL.debug=true`
- `watt.server.SoapRPC.debug=true`

BPM Products - Designer

The Software AG Designer is the Eclipse-based Development Environment used to design Business Process Models, Flow Services (from version 7.2 and higher) and Composite Applications.

Common Problems

Empty error message when regenerating a Web Service Connector (WSC)

There are a few causes for this issue, one of which is having applied the latest Designer_CAD* fix without installing the required / associated My webMethods Server Common Library fix.

Unable to build a CAF (Composite Application Framework) application

Ensure all the needed libraries are in the build path. Check whether the source files have been somehow renamed in the project folder.

Unable to build and upload a process model. Designer freezes and then the build process fails.

Check the JDBC Pool for Process Engine in Designer (or on the Integration Server if it is configured to use that one) and increase the idle timeout.

The image visualized in My webMethods Server has different icons for referenced process steps.

Add the following extended setting to your Integration Server:

- `watt.server.auditStore=database`

This extended setting was theoretically removed beginning in version 7.1.1. However, it is still used by Task Engine, Process Engine, and Monitor in order to display the process instance information (image, step details, etc.).

General recommendations

- Review the latest available fixes and install the ones that apply. Carefully check the fix dependencies (the Designer_CAD* fixes should always be installed with the My webMethods Server Common Library fixes if even there is no My webMethods Server running on the machine).
- Check for any errors in the logs. If this does not help, start Designer in debug mode and see if there are errors in the console.
For more information look at the section "Log File Associated with the Problem Component" below.

Information to Gather When Opening a Support Incident

- Designer release version
- [Log File Associated with the Problem Component](#)
- [List of Installed Fixes](#)
- [Configuration Details](#)

Configuration Details

In Designer, in the upper menu, navigate to Help > About webMethods Designer > Configuration Details. Select the Copy to Clipboard button to copy all of this information to the clipboard. Then paste it into a text file.

List of Installed Fixes

To determine the installed fixes:

- Navigate to WM_INSTALL_FOLDER/Designer/eclipse/plugins
- Look at the file names.
 - If any fix is installed, some file names end with the fix name
For example: "com.webmethods.caf.common_7.1.2.v93_DES_7.1.2_CAD_Fix7"
"com.webmethods.process.generator_7.1.2.v98_DES_Process_7-1-2_Fix5"

Log File Associated with the Problem Component

- The Designer log file is located in "<webMethods_install_directory>\.metadata\log".

Running Designer in Debug Mode

In order to gather additional troubleshooting information, it is possible to start the Software AG Designer in debug mode. When you start Designer in debug mode, the debug information is written to the console.

Start the Designer in debug mode as follows:

- `cd "<webMethods_install_directory>\Designer\eclipse"`
- `eclipse -vm <webMethods_install_directory>\jvm\win150\bin\java.exe -Dderby.storage.pageSize=400000 -product com.webmethods.product.designer.ide -consoleLog -debug`

BPM Products - Monitor

Common Problems

Archiving

There are two common archiving issues reported:

1. Archiving is not working.
2. Archiving not archiving all the data.

Archiving is not working

Please make sure the following steps have been completed if you find that Archiving is not working.

If the Archive Schema and Integration Server Core Audit Schema are installed on different Schemas

1. Create a separate Oracle schema dedicated to archiving.
2. Install the archive component using a separate Oracle schema dedicated to archiving.
3. Install the `ISCoreAudit` & `ProcessAudit` components using separate Oracle accounts dedicated to archiving.
4. Grant select & delete access for all tables to the Oracle schema dedicated to archiving.
5. Update the archive JDBC-Pool on the source Integration Server with a separate Oracle account dedicated to archiving connection info.
6. Update the `OPERATION_PARAMETER` table invoking service `pub.monitor.archive:setOperationParameters` from the source Integration Server. Please refer to the Monitor Built In Services Reference guide for usage instructions.
7. Invoke the archive service in the `pub.monitor.archive` package to perform archiving.

Please note that you don't have to individually install the components described step 2 and 3. When you choose the Archive option in the Database Component Configurator it should install Archive, Operation Management, IS Core Audit and Process Audit. Also, please make sure you have the latest Database Component Configurator installed.

If the Archive schema and IS Core Audit schema are installed on same schemes

If you have installed both the IS Core Audit and Archive components on the same schema, install the `ISCoreAudit` & `ProcessAudit` components and the Archive component using the Oracle schema.

While doing this, if you have installed `ISCoreAudit` and `ProcessAudit` before Archive, the database trigger may become disabled. This database trigger is used in the archive process and hence that can lead to archiving not working.

You might see an error like the following:

```
[MON.0119.0305E] Error in deleting driver tables and committing:
com.wm.monitor.util.MonitorException: Error in service archiving: {0}
2009-04-28 15:34:16 EST [MOD.0120.0762E] Error in service archiving: java.sql.SQLException: [wm-
cjdbc33-0079][Oracle JDBC Driver][Oracle]ORA-04080: trigger 'WMSERVICE_RAIU' does not exist
ORA-06512: at "ARCHIVE.SERVICE_ARCHIVE", line 96
ORA-06512: at line 1
```

Therefore, please make sure the database triggers WMPRCSSSSC_RAIU (WMPROCESSASSOC table), WMPRCSS_RAIU (WMPROCESS table) and WMSERVICE_RAIU (WMSERVICE table) exist and that they are enabled.

For example on an Oracle DB do the following:

```
select status from user_triggers where trigger_name like '%WMSERVICE_RAIU%';
```

If the status is Disabled, they should be enabled.

Archiving not archiving all the data

If the Archiving function does not archive all of the data, there is generally a simple explanation. It is possible that the user is only archiving 'Completed' data instead of 'Completed', 'Failed', 'Stopped' and 'Resubmitted' data, although this is clearly marked in Monitor. Most likely there are process instances or services in the tables that are still in the 'Started' status. Archiving will naturally not remove anything that is still 'Started', but also there could be entries in the error log that are related to such processes or services. These will also not be archived, unless the 'owning' service or process is archived first. You can check this either via the Monitor pages themselves (just look for old process instances or services in a 'Started' state) or on the database itself, look for entries in WMPROCESS and WMSERVICE tables where the 'STATUS' is '1'.

In order to correct this problem, see also ["Manually Set Process Instance Status to Completed, Stopped, or Failed"](#). Process instance status can be changed using the following Monitor service: `pub.monitor.process.instanceControl:changeInstanceStatus`. A similar service does not currently exist for services. Instead, incorrect WMSERVICE entries can only be removed either directly on the database, or by (again, directly on the database) setting the status of started services to '2' (completed) and then running archiving again.

Process Model

Can't Enable Process Model

The following errors can occur when attempting to enable a process model:

```
[MON.0119.1501D] Error checking portal privilege: java.lang.NullPointerException
```

We often see the following error, when trying to enable a process model:

```
2006-01-12 00:14:47 CET [MON.0119.1501D] Error checking portal privilege:
java.lang.NullPointerException
at electric.util.license.License.isInternalLicense(Unknown Source)
at electric.registry.Registry.getReference(Unknown Source)
at electric.registry.Registry.getProxy(Unknown Source)
```

This error is caused by having older Glue, Fabric or ServiceNet jar files still defined in the system classpath. The Integration Server About page - Server classpath provides details about the additional jars being included in the system classpath. Cleaning the system classpath and rebooting the machine generally resolves this error.

See also [Process Engine Troubleshooting - Cannot Enable Model](#).

How can I delete process instances with a status of "Started"

If you want to delete process instances with a status of "Started", first you need to stop them.

Stopping a Process Instance

You stop a process from the Monitoring > Business > Process Instances > Process Instance Detail page.

1. Display detailed information for the process instance that you want to stop.
2. On the Process Instance Information tab on the Process Instance Detail page, click Stop.

Clean up hose instances

After the instance is stopped, in order to clean up those instances, as described in "webMethods Monitor User's Guide" you have three options:

1. Archiving or Deleting Logged Data from My webMethods
2. Archiving or Deleting Logged Data using Built-in Services
3. Archiving or Deleting Logged Data using Stored Procedures

In particular, you can use the following Built-in service:

`pub.monitor.archive:processArchive`

This service archives or deletes process data and control data (such as resubmit actions) from the `WMCONTROL`, `WMCUSTOMFIELDDEFINITION`, `WMCUSTOMPROCESSDATA`, `WMPROCESS`, `WMPROCESSASSOC`, `WMPROCESSDEFINITION`, `WMPROCESSIMAGE`, `WMPROCESSRECENT`, `WMPROCESSSTEP`, `WMSTEPDEFINITION`, and `WMSTEPTRANSITIONDEFINITION` tables in the Process Audit Log database component, and archives or deletes process related service data from the `WMSERVICEACTIVITYLOG`, `WMSERVICE`, and `PPWMERROR}}` tables in the IS Core Audit Log database component.

Specifically, the service archives or deletes the following:

- Process log entries, input pipelines, error data, and runtime values for user specified input and output document fields.
- Referenced processes, process related service data (services, service error data, and user defined messages).
- Process control data (resubmit, suspend, and resume actions).

Service

Integration Monitoring cannot display the Audit services.

First make sure there are no errors displayed when you click on Integration Monitoring. Any exceptions/errors could be a result of improper configuration.

If there are no errors, and the services being audited are still not shown in Integration Monitoring services, then there could be a different issue.

Make sure the `WMSERVICE` table is populated with records when you execute an Integration Server service that is auditing enabled.

When you perform archive or delete operations on the Integration Server core audit data, Integration Server calls a database stored procedure and connects to the Integration Server database by using the connection configured in IS JDBC pool. The stored procedure will copy (or delete) the core audit data. Integration Server will then connect to the archiving database by using the connection configured in JDBC pool (Archiving functional alias) and insert the data to the archiving database/schema (if you choose to archive data).

Since the Process and Archiving database schemas are different entities, you need to configure separate JDBC pool aliases to point to the different database schemas.

If you are using the same JDBC pool for both Integration Server and Archiving, the stored procedure will actually disable the database triggers described in the following paragraphs.

Depending on whether you are archiving IS Core Audit or Process Audit, the stored procedure will disable the triggers accordingly.

For Service Archive, the stored procedure disables the trigger, `WMSERVICE_RAIU` (`WMSERVICE` table).

This trigger will actually copy the Service data from the `WMSERVICE` table to another related service data table called `WMSERVICEASSOC` which is used by MWS to display the audited services. Thus when this trigger is disabled, service data is not copied to the related table, and this could be one reason that you are unable to view the services under Integration Monitoring.

For Process Archive, the stored procedure disables the following three triggers:

- `WMPRCSSSSC_RAIU` (`WMPROCESSASSOC` table)
- `WMPCSS_RAIU` (`WMPROCESS` table)
- `WMSERVICE_RAIU` (`WMSERVICE` table)

The three triggers will actually copy process data to another related process data which are used to view process instances in MWS when a new process instance is created. Thus when the triggers are disabled, process data cannot be copied to other related process tables and this is why you are unable to view process instances in MWS.

A workaround to view new process instances in MWS is to re-enable the following triggers in the IS database:

- `WMPRCSSSSC_RAIU` (`WMPROCESSASSOC` table) - (For Process Instances monitoring issue)
- `WMPCSS_RAIU` (`WMPROCESS` table) - (For Process Instances monitoring issue)
- `WMSERVICE_RAIU` (`WMSERVICE` table) - (For both Process Instances and Services monitoring issue)

Executing Monitor built-in service throws a ClassCastException

Running the service WmMonitor/pub.monitor.process.model:getModelList in 8.0.1 throws the following exception:

```
$errorInfo $errorDump com.wm.monitor.util.MonitorException:
Could not retrieve model list: java.lang.ClassCastException: java.lang.Long
cannot be cast to java.sql.Timestamp at com.wm.monitor.app.Model.getModelList(Model.java:746) at
pub.monitor.process.model.getModelList
```

This is a known issue in Monitor 8.0. Please apply MON_8.0_SP1_Fix1 (or latest) to resolve this issue.

Can't Resubmit a Service

The following problems might occur when resubmitting a service:

When resubmitting several failed instances with MWS the limit of the max connections parameter in the JDBC Pool alias defined for the processAudit is a constraint. If n is the maximum number of connections, it is possible to resubmit n-2 process instances. Resubmitting more than this can consume the JDBC connections.

If you try to resubmit only one instance more you get the following error:

```
2008-03-25 18:10:39 CET [ISC.0088.0001E] SOAPException: [ISS.0088.9134] Exception occurred while
processing the body of the message
2008-03-25 18:10:42 CET [MON.0119.0042E] Failed in trying to resubmit all: java.sql.SQLException:
no connection returned from pool [ProcessAudit]
2008-03-25 18:10:42 CET [MOD.0120.0000D] java.sql.SQLException: no connection returned from pool
[ProcessAudit]
```

Monitor 7.0 SP2 Fix7, Monitor 7.1 Fix3, or Monitor 7.1.2 Fix1 (or latest) will help to resolve the issue by releasing the connection correctly.

General recommendations

Document Logging not working in 8.x

Check the following if you find that document logging is not working:

- IS is configured with JDBC Pools for IS Core Audit.
- IS is configured with Messaging > Broker settings.
- WmLogUtil package is enabled and Running.
- Document Type Logging is enabled in the Broker. In MWS Admin > Messaging > Broker Servers > Servers click on the Broker in the list where you have this document type defined. You will see a check box with label Document type logging. Select this check box.
- In MWS Admin > Messaging > Client Groups > Integration Server > Log Publish pick the document type and add it to this list.
- Check the Last Published column for that document type to make sure it has the correct timestamp when the document was last published to the Broker.

This confirms that the document type gets published to Broker fine, and only auditing is an issue.

- While configuring Document Logging, there will be two clients created by name ISClientPrefix_LogOnPublish and ISClientPrefix_LogOnAck. Please go to ISClientPrefix_LogOnPublish
Go to the Subscriptions tab, where you will see the list of Document types for that Broker. Select the document type that you would like to log, and click the Add Subscription button.

Issues in Service Audit/Process Audit

At times, you see the service/process auditing works for a while and suddenly stops auditing further in 8.0.1

This is a known issue. Please apply the IS_8.0_SP1_Core_Fix5 (or latest) to resolve this issue.

If the issue still persists, please contact Software AG Global Support.

Recovering After Network or Database Outages

If webMethods Monitor loses its connection to the process log database (for example, you restart the database or you experience a network failure), then Monitor may not be able to automatically begin displaying new activity when the connection returns. This occurs because failures like these break the connections that Monitor has established in the connection pool.

To correct this condition, you have to re-establish the connections by restarting the JDBC pools and reloading the Monitor package. To reload the WmMonitor package, open the Integration Server Administrator, select Management in the Packages menu, and click the Reload icon next to WmMonitor. To restart the JDBC Pools, open the Integration Server Administrator, select JDBC Pools under Settings menu. Then use the Restart menu for the IScoreAudit and ProcessAudit pools.

Information to Provide When Opening a Support Incident

When you submit a support incident related to an Integration Server problem, please provide the following information to Software AG Global Support.

List of Installed Fixes

Obtain the list of installed Monitor fixes from the Integration Server About page.

Note! - Not all fixes and service packs display on this page after they have been installed. Also check the "updates" link on the bottom of the about-page.

- The best practice is to provide screen shots of the following:
 - IS About page
 - IS updates page

Diagnostic Information

Because the nature of every problem can be different, so can the information required to troubleshoot and analyze the problem. However, the following information is typically requested for most support incidents submitted regarding Integration Server:

- **All Error Messages and/or Symptoms Witnessed**
Please provide complete error messages, including full stack traces (if any). There's a check box to see the stack traces in the error-log screen. Please copy and paste the entire error message text. Because Software AG Global Support staff uses the exact text to search for previous reports of the same error text, avoid abbreviating information or introducing typos when copying the error messages.
- **Integration Server Logs**
These logs are stored in either the Integration Server/logs directory or the database. The Integration Server logs related to Monitor include the following:
 - error log
 - server log
 - In order to obtain useful server log entries, set the server debug level to 4 (Debug) or higher (Trace). Higher debug levels are preferable if your server load, performance requirements, and available file system space can accommodate them.
 - If you set the debug levels high (6 or above, called 'trace' in IS 7.1), selectively disable log facilities that generate many log messages but that are irrelevant to the problem. The log facilities we typically ask you to disable are the following:
 - 0019 Monitor
 - 0020 Monitor Database Layer
- **Error messages displayed in the MWS**
If possible, provide a screen shot of the error(s) and MWS full.log and problems.log under MWS/server/default/logs, where default is the node name

BPM Products - My webMethods Server

Common Problems / Typical Causes

POP.005.0006 This server node is not defined in the My webMethods Server cluster configuration

This topic shows two ways to get My webMethods Server working with a new schema. We often recommend that one of these solutions be applied to a problematic environment.

Solution 1

- Install a new server instance pointing to a new database
- Run this command from the MWS/bin directory:

```
mws.bat -s <newinstancename> getconfig cluster.xml
```

- This downloads the cluster.xml file to the "<webMethods_install_directory>\MWS\server\<instance_name>\config" directory.
- Modify cluster.xml and change the node name. You can get the node name from the server.properties.bat file.
- Publish the cluster.xml file back to the database:

```
mws.bat -s <newinstancename> putconfig cluster.xml
```

- Delete cluster.xml from the config directory.
- Start My webMethods Server.

Solution 2

- You can copy the whole My webMethods Server installation to a different machine to the same "<webMethods_install_directory>\\" and reconfigure it to listen on a different port.

Unable to Add New Drivers for Custom Data Source in 8.2

For configuring a Custom Data Source in My webMethods Server 8.2, before creating a customized data source, you must ensure that you have the appropriate drivers available to the My webMethods Server.

Please follow the instructions in the Administering My webMethods Server guide, "Adding a Custom Data Source". In addition to this, you may want to run the following command before restarting the MWS server.

```
<webMethods_install_directory>\MWS\bin\mws.bat update
```

Note: there is a caching issue when using a Custom Data Source.

If you have entered an invalid database "Connection URL" while adding the new data source for the custom database, any further creation of new data sources will fail even with a valid database connection URL. This is because the host name will be cached in My webMethods Server by design. To overcome this situation, restart My webMethods Server to clear the cache and re-enter the valid database connection URL.

<MWS>\server\default\temp\ directory is huge

The files in the temp directory are created when you develop, deploy and run portlets and web applications on the My webMethods Server instance. When My webMethods Server starts, Jetty creates directories for each application and some temporary files are stored there. The files in the <MWS>\server\default\temp\ directory can be safely deleted when the My webMethods Server is down.

Performance and visualization problems with the My webMethods Server user interface when using Microsoft Internet Explorer

- Make sure Internet Explorer has the latest patches available from Microsoft installed.
- Install the latest My webMethods Server Fix available (for 7.1.2 a lot of improvements have been added in fix3 and later).

No implementation for auth scheme: "specifieduser".

The My webMethods Preview Server has an authentication handler component called "wm_specifieduserauth", that automatically logs in a user "Designer". Sometimes this component gets deployed to the full My webMethods Server installation. The result is that normal users cannot login and get the error message

```
Hello, Designer ! [POP.003.0081] No implementation for auth scheme: "specifieduser".
```

To resolve this problem

1. Shutdown My webMethods Server (if it is running)
2. Take a backup of the file "wm_specifieduserauth.cdp" under <MWS>\server\default\deploy and keep it outside of webMethods installation directory.
3. Remove the file "wm_specifieduserauth.cdp"
4. Restart the My webMethods Server

Current Connections are higher than the Max Connections

The number of current connections sometimes exceeds the maximum number of connections.

The following message is logged in this situation:

```
[CommonLib.CDS.0002W] CurrentConnections = 24; maxConnections = 20
```

This issue is fixed with MWS_8.0_SP2_Fix4 or the latest which supersedes Fix4.

Unable to create new ldap connection: Maximum number of connections exceeded.

When there are multiple concurrent requests to My webMethods Server from users defined in LDAP, the following error can occur.

```
[CommonLib.MWS.0002W] DB Server disconnected. Will try to reconnect
at com.wm.app.b2b.server.scheduler.ScheduleDB$ScheduledService.run(ScheduleDB.java:2733)
at com.wm.app.b2b.server.scheduler.ScheduleDB$ScheduledService.run(ScheduleDB.java:2733)
```

followed by

```
[CommonLib.CDS.0002W] Unable to create new ldap connection: Maximum number of connections
exceeded.
```

It is recommended to increase the maximum number of connections for LDAP from 20 (default) to a higher limit to handle all user requests.

It has been identified that My webMethods Server does not recycle the used LDAP connections correctly. This has been addressed in the following fixes.

- For MWS 8.0.2, apply MWS_8.0_SP2_Fix6 or higher.
- For MWS 7.1.2, apply MWS_7.1.2_Fix14 or higher.

MWS hangs with java.lang.OutOfMemoryError: Java heap space

My webMethods Server hangs after a few hours / days of normal use and becomes totally unresponsive. The error received is:

```
2011-06-20 23:03:33 EDT (org.mortbay.log:ERROR) - EXCEPTION
java.lang.OutOfMemoryError: Java heap space
```

This error typically indicates that the server is out of java heap space. Increasing the My webMethods Server maximum heap size should resolve this issue. The maximum heap size can be changed in the <MWS>\server\default\bin\server.properties.bat file for the server instance. The default property is:

```
JVM_OPTS=-Xms32m -Xmx1024m
```

You can increase the -Xmx value to a higher value that will be sufficient for your environment. Also increasing the Xms value will give a better performance of the My webMethods Server, because the JVM will not have to allocate more heap space until this value is exceeded.

General recommendations

- Always install the latest fixes available
 - Pay extra attention to the dependencies between fixes, because some fixes may be applicable to several products.
For example: Some Common Libraries (CL) fixes address problems in My webMethods Server as well as Integration Server. Therefore, dependencies may exist between a CL fix and My webMethods Server fixes as well as between that CL fix and Integration Server fixes. Also, if you have installed the servers on separate machines, the Common Libraries fix will need to be installed in both locations.
- If you are using My webMethods Server 8.2, for general troubleshooting of performance issues, please refer to the Diagnosing My webMethods Server documentation at http://documentation.softwareag.com/webmethods/wmsuites/wmsuite8-2_ga/My_webMethods/8-2-SP1_Diagnosing_My_webMethods_Server.pdf

Information to Gather When Opening a Support Incident

Version Information and Installed Fixes

To determine the My webMethods Server version and which fixes have been installed:

- Log in as the Administrator user.
- Click the "About" link in the upper right-hand corner.
- This page lists the My webMethods Server version.
- Service packs and fixes are listed on the "Licensed Products" line on the "My webMethods Server" and "Infrastructure" tabs.

Environment Details about Where the Problem is Occurring

- Version and name of the OS (and Service Pack if applicable).
- Name and version of the browser used.

Log File Associated with the Problem Component

The logs for the My webMethods Server instance can be found in
 "<webMethods_install_directory>\MWS\server\<instance_name>\logs".

In order to get more detailed logs you can log in as sysadmin and increase the logging levels for the facilities under:

Folders > Administrative Folders > Administration Dashboard > Analysis > Logging Configuration

You can also start My webMethods Server in debug mode as follows:

```
<webMethods_install_directory>\MWS\server\<instance_name>\bin\run.bat -d
```

"instance_name" is the instance name of the My webMethods Server. For the default instance, this is "default".

Running My webMethods Server in debug mode will record debug data in "full.log". If My webMethods Server is started in debug mode from a console it will echo the debug data to the console as well.

BPM Products - Process Engine

Introduction

The webMethods Process Engine, or PE (previously known as Process Runtime, or PRT) is a component of webMethods Integration Server that controls the run-time execution of business processes designed in Software AG Designer. It is installed with the Software AG Installer and appears in Integration Server as the package WmPRT. The Process Engine must be installed on every Integration Server that is used to run your business processes. Prior to version 7.x, process models are designed using webMethods Modeler.

New in 8.0.1:

- In 8.0.1, the Process Engine dashboard was redesigned and now displays some basic troubleshooting information. For example, startup log messages are displayed there. Additionally the status of various PE components is shown. For more information see Chapter 4 of the 8.0.1 PDF publication *Administering the Process Engine*.

New in 7.1.x:

- The Process Engine requires you to create a non-transactional JMS connection alias with the specific name `PE_NONTRANSACTIONAL_ALIAS` in the Integration Server.
For more information about creating a JMS connection alias and working with JMS triggers, see *Using webMethods Integration Server to Build a Client for JMS*.
- When you generate a process model in Software AG Designer, Designer automatically creates a .process file in the project directory of Designer's workspace directory, which is used to store the design time process data. Prior to version 7.x, webMethods Modeler stores process data in the Repov3.

The following items can help you troubleshoot Process Engine problems.

Common Problems / Typical Causes

Cannot Enable Model

The best way to determine whether a process model is properly enabled is to set the Integration Server logging to Trace or higher for facility 0119. See [Process Engine Trace Logs](#) for further details on Process Engine logging.

The following message appears in the server log when a process model is enabled:

```
[MON.0119.0076V4] Enabling process with IData control object:
>>>BasicData:EnableModelID=xxx,DisableTrackingModelID=xxx<<<
```

Join Condition Not Met

The following symptoms indicate that a join condition is not met:

- The process seems to have performed all required transitions to a step.
- Step remains in "Waiting" status. webMethods [Monitor] indicates this status by displaying an hourglass icon next to the step.

The best way to troubleshoot a join is to set the Integration Server logging level to Trace for facility 101 in the logging settings. See [Process Engine Trace Logs](#) for further details on Process Engine logging.

Multiple types of joins exist within Process Engine, and every join that is evaluated is expected to write specific information to the server log, as follows:

webMethods Broker Document Subscription

A join is evaluated for any Broker document subscription in the process model. In the following example, there is one join condition -- the subscription of the SimpleProc.Documents:Document:

```
(T4) join string:
(T4) join state: {}
(T4) join result: (jr (m SimpleProc.Documents:Document))
(T4) matched 1 events!
(T4) returning positive join result
```

webMethods Broker Document Subscription + Transition + Correlation

A join is evaluated for any Broker document subscription in the process model with a transition from a previous step. For example:

```
(T2) 1 events in the queue
(T2) event names: [TX-FROM-N2]
(T2) join string: (and SimpleProc.Documents:Document2 TX-FROM-N2)
(T2) join state: {}
(T2) join result: (jr)
-----
(T2) nothing to do for this step yet
(T2) ending synchronous execution of step N3
```

In this example:

- The join string has the following structure: **(and SimpleProc.Documents:Document2 TX-FROM-N2)**. The first part of the statement is the type of join (in this case, "and") and the second part of the join includes all the required events that must occur for the join evaluation to complete successfully.
- The log entry **event names: [TX-FROM-N2]** indicates that the transition from the previous step N2 has already been received. The step properties of the previous step indicate that the Unique ID of the previous step is N2.
- The last entry of the log also indicates that the step N3 cannot be executed yet. You may want to check the database to see if the join entry has been logged. To do so, check the database table WMPRTXREF to see all correlation data. Look for an EXTERNALID entry (the correlation ID) and an INSTANCEID (the instance ID of the process model); if these entries do not exist, there may be a problem with your correlation service generating a proper correlation ID.

Completed Join (Broker)

For every completed join involving a Broker document, the log file is expected to contain the event names associated with the completed join. For example:

```
(T2) 2 events in the queue
(T2) event names: [SimpleProc.Documents:Document2, TX-FROM-N2]
(T2) join string: (and SimpleProc.Documents:Document2 TX-FROM-N2)
(T2) join state: {}
(T2) join result: (jr (m SimpleProc.Documents:Document2 TX-FROM-N2))
(T2) matched 2 events!
(T2) returning positive join result
```

In the join result, the "m" in **(jr (m SimpleProc.Documents:Document2 TX-FROM-N2))** indicates a match of two events.

Incomplete Join (Broker)

For an incomplete join, the log file is expected to contain only the document name in the event names, without a join result match. For example:

```
(T2) 1 events in the queue
(T2) event names: [SimpleProc.Documents:Document2]
(T2) join string: (and SimpleProc.Documents:Document2 TX-FROM-N2)
(T2) join state: {}
(T2) join result: (jr)
-----
(T2) nothing to do for this step yet
(T2) ending synchronous execution of step N3
```

Trading Networks Document Subscription

A join is evaluated for any Trading Network subscription in the process model.

In the following example, the only join condition is a subscription of a Trading Networks document:

```
(T2) PRT queue processing for step N1
(T2) process instance ID: dd064f000d291ldb97f488cbac5596bb:1
(T2) 1 events in the queue
(T2) event names: [m1a6ce001bc4q4q30000001e]
(T2) join string:
(T2) join state: {}
(T2) join result: (jr (m m1a6ce001bc4q4q30000001e))
(T2) matched 1 events!
(T2) returning positive join result
```

The event name m1a6ce001bc4q4q30000001e refers to the TN document internal ID. You can find the TN document internal ID by clicking **Document Details** on the Trading Networks Transaction Analysis screen.

Trading Networks Document Subscription + Transition + Correlation

A join is evaluated for any Trading Networks document subscription in of a process model with a transition from a previous step. For example:

```
(T2) 1 events in the queue
(T2) event names: [TX-FROM-N2]
(T2) join string: (and m1a6ce001bc4q4q30000001e TX-FROM-N2)
(T2) join state: {}
(T2) join result: (jr)
-----
(T2) nothing to do for this step yet
(T2) ending synchronous execution of step N3
```

In this example:

- The join string now has the following structure: **(and m1a6ce001bc4q4q30000001e TX-FROM-N2)** . The first part of the statement is the type of join (in this case, "and") and the second part of the join includes all the required events that must occur for the join evaluation to complete successfully.

- The log entry **event names: [TX-FROM-N2]** indicates that the transition from the previous step N2 has already been received. The step properties of the previous step indicate that the Unique ID of the previous step is N2.
- The last entry of the log also indicates that the step N3 cannot be executed yet. You may want to check the database to see if the join entry has been logged. Check the database table WMPRTXREF to see all correlation data. Look for an EXTERNALID entry (the correlation ID) and an INSTANCEID (the conversation ID of the TN document); if these entries do not exist, there may be a problem with your correlation service generating a proper correlation ID.

Completed Join (TN)

For every completed join involving a TN document, the log file is expected to contain the event names associated with the completed join. For example:

```
(T2) 2 events in the queue
(T2) event names: [mla6ce001bc4q4q30000001e, TX-FROM-N2]
(T2) join string: (and mla6ce001bc4q4q30000001e TX-FROM-N2)
(T2) join state: {}
(T2) join result: (jr (m mla6ce001bc4q4q30000001e TX-FROM-N2))
(T2) matched 2 events!
(T2) returning positive join result
```

In the join result, the "m" in **(jr (m SimpleProc.Documents:Document2 TX-FROM-N2))** indicates a match of two events.

Incomplete Join (TN)

For an incomplete join, the log file is expected to contain only the document name in the event names, without a join result match. For example:

```
(T2) 1 events in the queue
(T2) event names: [mla6ce001bc4q4q30000001e]
(T2) join string: (and mla6ce001bc4q4q30000001e TX-FROM-N2)
(T2) join state: {}
(T2) join result: (jr)
-----
(T2) nothing to do for this step yet
(T2) ending synchronous execution of step N3
```

ServiceUtils.rethrow() Exception

You may see the following exception in the Integration Server server.log and/or error logs:

```
com.wm.app.b2b.server.ServiceException: [PRT.0101.9125] Service
wm.prt.dispatch:handlePublishedInput failed
at com.wm.app.prt.util.ServiceUtils.rethrow(ServiceUtils.java:57)
at wm.prt.dispatch.handlePublishedInput(dispatch.java:136)
...
```

This exception is thrown by code which catches an exception and re-throws it. The underlying exception is not obvious and can occur for various reasons.

Process Instance Does Not Start

Check the following:

- Ensure that the process model is **enabled** in Monitor.
- If the first step of your process subscribes to a **Trading Networks** document type, make sure the TN document type extracts the ConversationID attribute.
- If the first step of your process subscribes to a document that comes from Broker (that is, a document that is published by another Broker client), use Software AG Designer (or webMethods Developer for v6.x and lower) to verify that the document type has been defined and that its Publishable property is set to **true**.

This issue may also result when Process Engine trigger threads stop responding (or "hang"). Obtain a thread dump to investigate the cause of the trigger hang. For more information about obtaining a thread dump, see the [How To Obtain a JVM Thread Dump](#) article.

Processes With Loops

- If a process has a loop, be sure to drop the pipeline before transitioning back to the originating step. If this is not done, the pipeline will carry the data from the previous iteration, resulting in duplication. This is often mistaken for a defect in Process Engine.
- When calling a service from a parent service, Process Engine restores the pipeline to that of the parent service. Therefore, use a "drop pipeline" rather than a "clear pipeline" service.

Web Service Invocation Failures

The Integration Server SOAP stack treats all SOAP failures in the same manner. This can be confusing because a connection failure can appear to be the same as an authentication or service failure. Run the process with debug level 8 applied and examine the server log carefully to determine the actual cause of the failure.

Step Retries

- Trigger retries represents the maximum number of times a trigger will be retried per the Integration Server retry queue upon receiving an Integration Server runtime exception.
- Step retries represents the maximum number of times a step will be executed. To ensure the proper retry behavior for a step, set the Step Retries property for the step in the model to the same value as Trigger Retries in the corresponding flow.
- In Designer, avoid placing a retry transition on an error step. However, if you set Step Retries > 1, generation will fail without a retry transition, even on retries for an error step. Set Retries to 0 (no retry limit) on the error step to generate.

Multiple Server / Cluster

If you are running a process on multiple servers and volatile tracking is enabled, you must disable the Local Correlation property for the process. Failure to do so will prevent the correlation from being propagated from one server to the other.

Volatile tracking is not supported in a clustered environment.

Process Recovery after Integration Server Crash

With volatile tracking, all Process Engine transition states are saved in RAM and not in the Process Engine tables. Process Engine does not guarantee recoverability when volatile tracking is used.

Slow Processes

When non-volatile tracking is enabled, Process Engine writes transition states to the Process Engine tables. This database access can slow down the Process Engine. Check the following:

- Ensure that you have the latest [DataDirect driver](#) version.
- ProcessAudit is used by Process Engine and by Monitor for auditing purposes. Ensure that ProcessAudit has enough pool resources. Furthermore, ensure the IS Audit Subsystem is tuned appropriately.
- Ensure that the Process Engine related document triggers are tuned properly.
- By default, Process Engine will retry the execution of a process for 10000 ms. This default value may make it seem as if no activity is occurring.
- Check the size of the Process Engine tables PRTPROCESS, PRTQUEUECONTENT, and WMPRTXREF. Large tables can slow down processing. To reduce their size, you can reduce the database cleanup interval by editing the "Cleanup Service Execution Interval" setting on the homepage of the WmPRT package[PRT]. You can also increase performance by creating indexes for these tables as follows:
 - PRTQUEUECONTENT and WMPRTXREF: new index by INSTANCEID
 - PRTPROCESS: new index by STATUS

Prevent In-Memory Deadlock

Use either of the following procedures to disable in-memory syncing of suspended processes in Process Engine:

From a text editor:

1. Locate the `prt.cnf` file in the `..\IntegrationServer\packages\WmPRT\config` directory of your installation.
2. Add or modify the following setting in the `prt.cnf` file: `watt.prt.externalcluster=true`
3. Save the `prt.cnf` file.
4. Stop and restart Integration Server.
5. Reload the `WmPRT` package from the Integration Server Administrator.

From the Process Engine home page in the Integration Server Administrator:

1. Click `Packages > Management`.
2. Locate the `{{WmPRT}}` package and click the Home Page icon.
3. Click `Settings`.
4. On the `Settings` page, click the `Edit Process Engine Settings` link.
5. On the `Process Engine > Settings > Edit` page, set the `External Cluster` option.

This setting applies to both clustered and non-clustered environments, even though the setting name is "externalcluster." It is intended to be used any time you want the Process Engine to use the database for synchronization rather than memory (this is typically done in clusters of servers, hence the setting name).

Note that if this setting is enabled and there is no actual cluster, `PEInternalTopic` documents end up in the Dead Letter Queue (if enabled). There will be one document for each process instance. This can be problematic on a busy system. You can work around this problem by creating a trigger that subscribes to the document.

Integration Server Throws a "No JMS Connection Alias found with name PE_NONTRANSACTIONAL_ALIAS" Error

This information applies to errors thrown while changing the process instance status in a 7.1.1 cluster environment.

The JMS connection alias - **PE_NONTRANSACTIONAL_ALIAS** is required when you have Process Engine module enabled in the Integration Server and expect to run a process model in a cluster environment. If this alias is not defined, Integration Server will throw the following exception while changing the process instance status:

```
[PRT.0101.0158T] Sending wm.prt.status:ChangeCommand over JMS to PEInternalTopic
[PRT.0101.0159T] Sending >>>BasicData:ProcessInstanceID=xxx<<<
[ISS.0014.0016T] Invoking service pub.jms:send
[PRT.0101.0059T] *** Error sending wm.prt.status:ChangeCommand: [ISS.0134.9017] No JMS Connection
Alias found with name "PE_NONTRANSACTIONAL_ALIAS"
```

As a result, the process instance is terminated abnormally.

To resolve this issue, you must configure the **PE_NONTRANSACTIONAL_ALIAS** in Integration Server Administrator > Settings > Messaging > JMS Configuration > JMS Settings > Create JMS Connection Alias. For information about how to create a JMS connection alias, see "Creating a JMS Connection Alias" in *Using webMethods Integration Server to Build a Client for JMS*.

Problem with Non-Deterministic Pipeline Merging

This is a known limitation with steps that loop over the same data element. With each iteration of the loop, when the pipeline is created, there is currently no way to distinguish between the current and the old values and the result is therefore non-deterministic. All pipeline variables inside the steps that participate in a looping pattern need to be carefully examined and the step pipelines must be cleaned up whenever possible so as not to propagate old values.

For example:

1. Suppose there are 3 processes, P1, P2, and P3. P1 refers to P2 and P2 refers to P3 (P1->P2->P3).
2. In P1, on certain criteria (such as by checking the process instance's pipeline data), P1 calls P2 again (looping).
3. The first flow works as expected (P1->P2->P3->P2->P1).
4. However, this flow does not work if it loops multiple times (P1->P2->P3->P2->P1->P2).
5. After 3-4 iterations, the process model completes, even though it was given instructions to loop again.

To resolve this issue, you must clean up the pipeline data (drop the pipeline object in the flow service) before looping back to P1.

QoS for Referenced Models

When using referenced models, both the parent model and the referenced process model should use either volatile or non-volatile tracking. If volatile and non-volatile tracking is not used consistently by the parent process and the referenced / child model, Modeler or Designer will use different mechanisms to maintain process status information. As a result, the Process Engine will be unable to detect when the referenced process model ends.

Resubmission in a Referenced Process Model (Bottom-up) Fails to Propagate Back to the Parent Process Model (webMethods 7.1.x)

Scenario:

A parent process model A invokes a referenced (child) process model B. If a step in the child process fails, you resubmit the failed step in the child process (after fixing the run-time error in the child process).

The expected behavior might be that the child process would return its status (for example, completed) back to the parent process after resubmission (bottom-up resubmission) and then re-join the parent process.

However, this does not happen in webMethods 7.x. Even though the child process instance completes successfully after resubmission of the failed step, the transition does not join back to the parent process, nor does the completed status of the child process propagate back to the parent process. This results in the parent process status remaining as "FAILED" whereas the child process status is marked as "COMPLETED".

To resolve this issue in webMethods 7.x, you can perform top-down resubmission (that is, resubmit the reference process step at the parent process) as opposed to a bottom-up resubmission (resubmit the failed process step from the child process).

Alternatively, configure the child process so that the status goes to "Suspended" instead of "Failed" on a failure. In this case, the parent process will remain in "Started" status and will wait for the child process to be completed. After you fix the error at child process and resubmit the failed step, the process transition will join back the parent process.

Process Engine 8.x allows users to configure models at design time to not "escalate" failure statuses to a given parent model. In this case, a failure in a referenced process will result in a "failed" child model, meanwhile the parent will remain in the "started" state.

Checklist Before Opening a Support Incident

Be sure to carry out all of the items on this checklist before you contact Software AG Global Support to open a support incident:

- You have reviewed the issues outlined in the section [Common Problems / Typical Causes](#).
- Ensure the product is correctly installed; that is, all steps of installation checklist are followed.
- Ensure any third party libraries are at the correct version level.
- Ensure the correct support or service packs for any third party software are installed.
- Ensure the latest third party fixes are installed in correct order.
- Ensure all screen shots, logs, thread dumps, and other information as described below are collected.

Information to Gather When Opening a Support Incident

Please provide the following information when submitting a Process Engine-related Support Incident to Software AG Global Support:

Environment Details

- Operating system and version
- Available RAM
- CPU info
- Hard drive information
- Test or production system

Screen Captures of Key Information

Screen shot of the Integration Server Administrator About page to determine the installed fixes:

1. Open the Integration Server Administrator (typically located at [http://hostname:5555/|http://hostname:5555/]).
2. Click the "About" link in the upper right-hand corner.
3. Take a screen capture of the first part of this page, from the top to the end of the "Software" section.

Screen shot of the Integration Server About > Packages/Updates page:

1. In the Integration Server Administrator About page, scroll to the bottom.
2. In the entry "Packages/Updates," click on the View link.
3. On the Support > webMethods Packages and Updates page, take a screen capture of the entire page (may require two captures).
Alternatively, you can browse to http://IntegrationServer:Port/WmRoot/updates.dsp and capture the screen there.

If this issue relates to process monitoring in My webMethods Server, obtain a screen shot of the My webMethods Server About page:

1. Log into My webMethods Server using a browser.
2. Click the "About" link in the upper right-hand corner.
3. This page displays tabs for My webMethods Server and the currently installed components (for example, Monitor, Task Engine, Trading Networks etc.).
4. Capture a screen shot from each tab. If the tab has a "System Information" button, click it and take a screen shot of that page also.

Environment Specific Information

Record and provide the following information:

- Integration Server's java parameters, heap size, thread pool size setting etc. (in case of thread usage or memory-related errors).
- Process Audit and Process Engine JDBC functional alias setting.
- Environment implementation description; for example, whether it is a standalone or clustered/non-clustered/distributed Process Engine.
- Process model implementation description; for example, whether multiple physical servers are involved in the process.

Configuration File Associated with the Problem Component

Obtain and provide the following files:

- Process Engine configuration file in the `<IntegrationServer>\packages\WmPRT\config` directory:
 - prt.cnf
- If one or more specific process models are involved, provide the process fragment file(s) in the `<IntegrationServer>\packages\<Process_Package_Name>\config\wmpmt` directory:
 - `<Process_Name>.<Design_Server_Name>.xml`

Exported Process Model Archive File from Software AG Designer

If one or more specific process models are involved, provide the process project file(s) exported from Designer.

To export a process model from Software AG Designer:

1. Open Software AG Designer.
2. Click File > Export > Software AG > BPEL File or Process File.
3. Click Next.
4. Select the process project you want to export from.
5. Select the process model(s) you want to export.

6. Click Browse to specify a target location for saving the export file.
7. Click Finish.

If you need to import a process project into your Designer workspace:

1. Open Software AG Designer.
2. Click File > Import > Software AG > BPEL File or Process File
3. Browse to and select the process file.
4. Click Open.
5. Click Finish.

The Quality of Service (QoS) Settings

Provide the setting values for these QoS options:

- Volatile Tracking
- Volatile Transition Documents
- Optimize Locally
- Express Pipeline
- Logging Level

To obtain the QoS settings for a process model

1. Open the process model in the Process Development perspective of Designer.
2. In the Properties view, click the Run time tab.
3. Record the Quality Of Service settings.

To obtain the QoS setting for Process Engine:

1. Open the Integration Server Administrator.
2. Click Packages > Management.
3. Locate the WmPRT package and click the Home Page icon.
4. Click Settings.

For more information about setting Quality of Service for a Process, see the PDF publication, "Administering the webMethods Process Engine."

Process Engine Trace Logs

Provide the <IntegrationServer>\logs\server.log file with the Process Engine at trace level when the problem occurs.

Before you obtain the server.log file, first extend the Integration Server logging level to "Trace" for Process Engine (log facilities 98 and 101) and Monitor (log facilities 119, 120 and 121). Then recreate the problem to obtain more debugging information; for example, detailed process step transition log, error/warning message, and the stack trace.

To set the logging level:

1. Open the Integration Server Administrator.
2. Click Settings > Logging.
3. Select the respective facilities as noted above. For Broker/trigger issues, facilities 97 and 99 must also be enabled.
4. Change the logging level to "Trace" (for Integration Server version 6.x, set it to 8 or higher).
5. Save the changes.

Output of `wm.prt.debug:inspectDB`

Beginning with version 7.1 you can use the service `wm.prt.debug:inspectDB` (in the `WmPRT` package) to provide useful information about a given process instance.

- The service will capture details about the instance from various Process Engine tables.
- This service takes in two arguments:
 - `InstanceId` - the Instance ID of the process instance in question.
 - `InstanceIteration` - the iteration # of the process instance in question.
- The output of this service, along with with the Integration Server logs set to Trace, can provide a great deal of information about any problematic process instances. In cases where an Integration Server log set to Trace is not available, the output of this service may still provide useful information for determining a root cause.

To use the `wm.prt.debug:inspectDB` service, simply run the service using Developer or through the IS Admin -> Packages Management -> WmPRT -> Browse Services -> `wm.prt.debug:inspectDB` -> Test `inspectDB`

Note: This is an "internal" service, and should only be used for debugging or troubleshooting purposes.

Thread Dump

For issues where a process stops responding (that is, a "process hang"), it is necessary to obtain a series of Integration Server thread dumps (at least 3 - 5 dumps) *while the hang is happening and before Integration Server is restarted*. A thread dump can provide far more information about the runtime thread behavior if it is taken as the hang occurs.

For more information about obtaining a thread dump, see the [Thread Dump](#) section.

BPM Products - Task Engine

General Recommendations

Fixes

Software AG suggests reviewing the latest fixes, which are available on [Empower](#) (a user account is required). Because fixes are cumulative, readme files provide the details about issues that are fixed by previous fixes as well. Readme files also contain the information about the fix requirements and installation instructions.

Software AG strongly recommends installing the latest maintenance release. New fixes are provided only for the latest maintenance release. For example, assume that the current maintenance release of 7.1 is SP2 (that is, 7.1.2). In this case new fixes will be delivered only for 7.1.2; there will not be any new fixes created for 7.1.1.

Common Problems

The following sections contain symptoms and possible resolutions for common Task Engine problems.

Invalid Credential Message for Integration Server

Problem

This message appears on the following My webMethods page: Administration > My webMethods > System Settings, after clicking the Check Server Status button Under the Task Engine Settings: WARNING: Invalid Credentials. Please validate user information for Task Engine to connect to Integration Server.

Possible Cause/Resolution

Apply OPTPROC_7-1_SP2_Fix4 (or latest) to resolve this warning message.

Task Queueing

Problem

Tasks are not being queued as expected.

Possible Cause/Resolution

The following are some causes for this problem:

- Check to see if the problem happens to multiple task types or with just a specific task type. You can also capture the My webMethods Server logs full.log and problems.log found in `.../MWS/server/default/logs/` and monitor them to see if any errors are thrown at the time the task is supposed to be queued.
 - If the problem is happening with multiple or all task types, the problem is most likely environment/configuration related.
 - If the problem is happening with a specific task type, the problem is related to just that task type.
- Ensure the Administrator user password is the same for both My webMethods Server and Task Engine.
- Ensure that the My webMethods Server/Task Engine settings are correctly configured on the WmTaskClient home page in Integration Server Administrator. For more information, see the Task Engine User's Guide.
- Also, if you are expecting the task to be shown in the Task Type Inbox, make sure they also appear in the Task List Management results list.

E-mail Notifications

Problem

E-mail notifications are not being triggered.

Possible Cause/Resolution

Ensure that you followed these steps for e-mail notifications:

(Further information about these activities is available in the Task Engine User's Guide and in the Designer Task Development Online Help)

1. Create a task notification with the task editor in Designer. This creates a subscription view user interface for the notification.
2. Define an event in the task editor to trigger this notification.
3. Configure the SMTP email server details in My webMethods (Administration > My webMethods > Email Servers).
4. Ensure that a valid e-mail address exists in the My webMethods Server user profile for the user to whom these notifications will be sent.
5. Subscribe the user expecting the notification to the notification in My webMethods.

If notifications are still not working after checking all the above, capture the My webMethods Server logs (as described below) with Task Categories set to debug, and contact Global Support.

Information to Gather When Opening a Support Incident

Please provide the following information when submitting a Task Engine related Support Incident to Software AG Global Support.

Clear Description of WHAT is the Problem

- Provide a short description of the problem you are facing. For example, only specifying that you are seeing an error message is not enough. Apart from the error message, specify what is the side-effect, what is the problem?
- What is the business impact caused by the problem ?

Environment Details About WHERE the Problem Is Occurring

- General information about WHERE ?
 - Where is the problem occurring? Development/QA/Production ?
 - Are you able to reproduce the issue on any other environment?

Specify WHEN the Problem Started

- When was the first time you started seeing the problem?
- How long were you running in a stable mode before the problem started? Or was the problem always there?
- How often is the problem occurring since it started? Can you recognize any pattern?

Version Information

Obtain the following information by clicking the "About" link in the upper right-hand corner of My webMethods:

- My webMethods Server version
- Task Engine version

List of Installed Fixes

Create a list of installed fixes and updates:

1. Open a command prompt.
2. cd to the ../MWS/bin directory.
3. Run the following command:
 - For UNIX systems: mws.sh updatesinfo
 - For Windows systems: mws.bat updatesinfo

For additional information, run the command mws.sh ? or mws.bat ?

Diagnostic Information

As the nature of every problem can be different, so can the information required to troubleshoot and analyze the problem. To obtain useful log entries, you must set the My webMethods Server task facility log level to Trace (at a minimum). Higher debug levels are preferable if your server load, performance requirements, and available filesystem space can accommodate them. To set the Task logging facility set to Trace:

1. Login to My webMethods Server as sysadmin.
2. Navigate to Folders > Administrative Folders > Administration Dashboard > Analysis > Logging Configuration.
3. Select Trace for Task.

The following pieces of information are typically required for most Task Engine Support Incidents:

- **All Error Messages and/or Symptoms Witnessed**
Please provide complete error messages, including full stack traces (if any). Because Software AG Global Support staff uses the exact text to search for previous reports of the same error text, avoid abbreviating information or introducing typos when recording the error messages.
- **My webMethods Server logs**
Provide copies of these log files stored in the directory `..\MWS\server\default\logs`.
 - `full.log`
 - `problems.log`
 - `errors.xml`
- **A series of thread dumps** This is only relevant when Task Engine appears to be unresponsive ("hanging").
 - Make sure you also describe in what way the Task Engine appears to be hanging.

Please provide a series of 5 thread dumps, taken at least 10 seconds apart. This helps Support to determine which threads are progressing and which threads are not progressing. Based on the nature of the problem, we may ask for additional thread dumps at different intervals.

For more information about obtaining a thread dump, see the [Thread Dumps](#) section.

SOA Products - CentraSite

Diagnostics

Log file locations

For CentraSite Governance Edition (7.1)

The log file is located in

```
<CentraSite installation directory>/jboss-4.0.5.GA/server/default/log
```

For CentraSite Enterprise Edition (3.1.x)

On Windows, the log file is located in

```
<drive>:\Documents and Settings\All Users\Application Data\CentraSite\CentraSite
```

On UNIX, the log file is located in

```
/opt/centrasite/CentraSite
```

For CentraSite ActiveSOA (8.x.x)

The log file is located in

```
<SoftwareAG installation directory>/CentraSite/data
```

In this location you will find the nucleus log, as well as the dumps resulting from a nucleus crash. The name of the nucleus log file is AABinosrv.<timestamp>.txt, where timestamp reflects the time when the server is started. For example AABinosrv.09-13-59.txt. On Unix this will be AABinosrv.09:13:59.txt (Using another timestamp syntax).

Note that the server output file is removed if the server process finally exits with a success status code.

Besides the server output file described above, other output files of the same format may exist. These have a different process name, for example AABinodmp.10:15:28.txt, which contains the output of the inodmp utility (backup and restore operation), or AABinoidx.11:55:13.txt, which contains the output of the inoidx utility (database index handling). These files are created if such a utility is either started via an SMH interaction or by the CentraSite server itself. If the created processes exit with a success status code these files are also removed.

Apache Log file

The Apache installation directory can be found under

- Centrasite Enterprise Edition (3.1.x)
`<CentraSite installation directory>\CentraSite\apache (Windows)`
`<CentraSite installation directory>/CentraSite_<version>/apache (UNIX)`
- Centrasite ActiveSOA (8.x.x)
`<SoftwareAG installation directory>/CentraSite/apache`

The Apache configuration file "httpd.conf" can be found in the "conf" directory of the Apache installation. The CentraSite-specific configuration entries are found via the following include statement in "httpd.conf" and are located in the "webconf" directory of the CentraSite installation:

```
include ../webconf/CentraSite*.conf.
```

The Apache access and error log files are normally found in the "logs" directory of the Apache installation.

Common Problems and Typical Causes

Cannot login to Centrasite UI

- Connection handler `com.centrasite.control.login` failed to establish connection.
Reason: Database access error. Can't connect to server CentraSite.
Plug-in `com.centrasite.control` has been disabled.

This means that the Registry/Repository is not started. Go to SMH > Managed Hosts > CentraSite > Registry/Repository.

Right-Click and Start. If it already Started, restart it.

- Connection Refused.Password Manager Exception occurred while initializing UDDI' error

The authentication between SMH, Tomcat and CentraSite is managed by processes named "sagssxauth2".

There are 3 locations for the "sagssxauth2" files:

- `<SoftwareAG installation directory>/common/arg/bin`
- `<SoftwareAG installation directory>/cpt/v80/opt/native/bin`
- `<SoftwareAG installation directory>/CentraSite/bin`

Each one of these files must be owned by `root:sag` and must have the sticky bit set, resulting in the following file permissions:

```
-rwsr-xr-x 1 root sag 15016 nov 18 00:32 ./ctp/v80/opt/native/bin/sagssxauthd2
-rwsr-xr-x 1 root sag 15016 nov 4 00:06 ./CentraSite/bin/sagssxauthd2
-rwsr-xr-x 1 root sag 15016 nov 20 19:09 ./common/arg/bin/sagssxauthd2
```

To fix the ownership and sticky bit, you have to execute the following commands:

- `chown root sagssxauthd2`
- `chmod u+s sagssxauthd2`

Information to Gather When Opening a Support Incident

Please provide the following information when submitting a CentraSite-related support incident to Software AG Global Support.

Environment Details about Where the Problem is Occurring

- Operating System (Windows, Unix, Linux) and version, as well as its architecture (32 or 64 bits).

Product version and list of Installed Fixes

To determine the installed fixes in CentraSite:

- Open the Centrasite Control (typically located at `http://localhost:53307/PluggableUI/?PLUGIN=com.centrasite.control&LOCALE=en`).
- Click the "About" link in the upper right-hand corner.
- This page lists the Centrasite version, Build number and build date.

The version of the System Management Hub is also relevant, if an administration issue is being faced. To find out the version:

- Open the SMH page, usually at `http://localhost:10010/PluggableUI/servlet/StartCISPage?PAGEURL=/PluggableUI/Login.html&PLUGIN=com.softwareag.systemmanagementhub&LOCALE=en`
- Click the "About" link and provide all the returned information.

Log File Associated with the Problem Component

As described at the start of this section, there are several different logs which may be helpful for diagnosing a problem. However, because they have to be obtained from different components, Software AG Global Support will guide the customer through setting up the logging which is needed for the issue in question.

SOA Products - Tamino

Tamino Troubleshooting

This section lists information that needs to be gathered for problems encountered with different scenarios.

For example: when a Tamino database crashes, Global Support will always ask for log files, database dump files and either Unix core or Windows Dr. Watson dumps.

The complete error code INOcccnnnn and error text.

Legend of the error code: ccc = Character; nnnn = Digits

Scenarios

Scenario	Diagnostic means
Server crashed	<ul style="list-style-type: none"> Database log files dump files UNIX core or Windows Dr. Watson dump file
Server hanging	<ul style="list-style-type: none"> pstack (Solaris) can be used to obtain a call stack userdump (Windows) can be used to obtain a memory dump. inodba kill=dbname can be used to obtain dump files (datastore, SMP, XMachine)
Server looping	<ul style="list-style-type: none"> pstack (Solaris) can be used to obtain a call stack userdump (Windows) can be used to obtain a memory dump. truss (some UNIX versions) can be used to monitor system calls being executed when looping. inodba kill=dbname can be used to obtain dump files (datastore, SMP, XMachine)
Unexpected response from X-Machine command	<ul style="list-style-type: none"> XML request log Datastore command log Trap datastore response code Trap XMachine response code
Need to debug inosrv in SET VERSION (upgrade)	debug inosrv started implicitly
Need to debug utility started by inosrv	debug inoxxx started implicitly

Installation / De-installation

Questions:

- Are other Tamino versions already installed? If yes, which version of Base Technology (BTY) has been installed?
- Which version of System Management Hub (SMH) has been installed?
- Which directory server has been installed: XTS or ADI?
- Which webserver If Apache: from Tamino-CD?

Information needed:

- Inst.log
- Inst.dbg (UNIX)
- httpd.conf (Apache)
- INS v8 Enable logging in the installer and get log

Migration

Questions:

- Description of how the migration was done (step-by-step)
- Migration was started from which version?

Information needed:

- Joblog (inosrv.txt),
- Backup of the database

Database Crash

Questions:

- Which action crashes the database?
- Can the database be started again?
- Is the crash reproducible If yes: how?

Information needed:

- Joblog (***inosrv.txt) +
- mentioned dumps:
- Tamino.***.xxx.xml
- SAGSMP.A.001.xxx
- Win: Dr. Watson log and user.dump
- UNIX: if existent: core file

***=database key (e.g. AABinosrv.txt)

xxx=timestamp

Tamino Interactive Interface (TII) or Query

Questions:

- Which command is used?
- Error code/text: INOcccnnnn
- If Query: are you using X-Query or XQuery?

Information needed:

- schema +
- example data
- query;
- XML Request Log (if query + performance problems)

Loader Tools

Questions:

- Is the loader called via Tamino Manager or via command line?
- If command line: which Loader is used (inoxmld or inoload)?

Information needed:

- Complete command +
- mentioned logfile
- schema +
- example data to reproduce loader issue

System Management Hub (SMH)/ Tamino Manager

Questions:

- Which action is used, on which node has been clicked on? (start of SMH, start of the database)
- Is an error number shown? (ARG _ _ _ _)

Information needed:

- Joblog of done actions (can be found in job monitor)

Security

Questions:

- Which kind of authentication is used: Tamino or Webserver?
- Which user/password is used for this action?
- Error code: INOcccnnnn

Information needed:

- ino:security

API / JAVA API (API4J) / API for C

Questions:

- API4J: which Java-classes are used
Content of classpath and stacktrace
- API for C: Content of PATH and LD_LIBRARY_PATH (UNIX)
Access via webserver or webserverless access
- What happens if this is done via Tamino Interactive Interface (TII)?

Information needed:

- Error code
- Logfile
- http logfiles
- example program
- Java-API: classpath + stacktrace

Server Extension (SXS)

Questions:

- Which server extension is used?
- Error code and complete message: INOcccnnnn?
- Used Programming language (JAVA, C++)?

Information needed:

- Example program
- SXP-files

HA /Cluster

Questions:

- Is the HA Kit from Software AG being used or did you write own scripts?
- Does the problem also occur in non HA-environment?
- Which Cluster software (which version!) is used?

Information needed:

- Inosrv.txt
- Cluster log: engine_A.log
- Scripts

Tamino Schema Editor (TSD)

Questions:

- Which actions lead to this error (Define, Import etc.)?
- What is the complete error code and -text: INOcccnnnn?
- What is defined/imported: schema or DTD?

Information needed:

- complete error code
- schema and/or DTD

X-Plorer

Questions:

- Connecting to local of remote database?
- Action that causes error

WebDAV Server

Questions:

- Is a DTD loaded?
- Does the error also occur with Tamino Interactive Interface (TII)?

Information needed:

- schema and/or DTD

History

Questions:

- Provide references to previous occurrences of the problem, Support Incident and/or Support Issue numbers, workarounds, and resolution.
- e.g., SI 1234567 has a similar exception
- e.g., Workaround in SI 7654321 works for certain cases

Information needed:

- Related SR and/or SI log numbers

Other Symptoms

Examples:

- High memory usage.
- Signal error; e.g. signal-11

Information needed:

- Log (*.txt) file

Diagnostic Information

Questions:

- Document all attempts to reproduce the problem.
- Does the problem occur in all OS/databases/versions
- Clear understanding in which situation the error happened

Information needed:

- Repro steps + any required schemas/DTD, data, queries +
- Core files (UNIX) or Dr. Watson (Windows) + Server
- Logs

Environment Details about Where the Problem is Occurring

Environment

- Operating system type and version
- UNIX: `uname -a`
- Windows version & SP

Webserver

- Which webserver? If Apache: from Tamino-CD?
- Apache Web Server v1 ($\geq 2.0.x$) or v2 ($\geq 2.2.x$)?
- MS IIS/SUN ONE/iPlanet/Other (version?)
- Test: Output of `http://localhost` in the browser

Current Installed Fix

- What is the latest fix that has been installed?

SOA Products - Insight

Introduction

The most common issue when working with Insight is no activity is seen after you have brought the node under management. To troubleshoot this and other issues:

1. Check uplink.cfg
2. Use the -debug flag
3. Audit agent events
4. Is plugmaker properly hooked in?
5. Look for stack traces
6. Make sure AOP files are getting parsed
7. Generate a support.zip file
8. Remove all flags you inserted

1 - Make sure that uplink.cfg is created

Make sure that "Uplink.cfg" got created on the machine where the Agent is deployed. If it is not created, then the interceptors think that the analyzer is down and they won't send agent events. There are typically two reasons for "Uplink.cfg" to not show up:

- The analyzer is not properly brought under management. Usually, this happens when the analyzer's configuration is corrupted and there are old Agent Configurations lingering around. To fix the situation, go into the analyzer's admin console and look for a tab called "Server Maintenance". If you see a URL to a server that's not supposed to be in the list, then remove it.
- The analyzer does not have read/write access to the directory where "Uplink.cfg" is written. If this folder is not read/write enabled for the user, you will need to relocate the "Uplink.cfg".

2 - The -debug flag

You can put the interceptor into "debug" mode by adding the "-debug" flag to the call to "actional-instrument.cmd" in your container start-up .bat or .sh file.

Here is an example:

```
%ACTIONAL_HOME%\actional-instrument.cmd -all -uplinkDir C:\tmp -debug
```

This will cause the interceptor to report a lot of additional information as it works. Note that this information will be contained in the startup log file for your container. You can put the results of the -debug flag into a separate file by adding the "-log" parameter to the startup string. Here is another example with the -log parameter added:

```
%ACTIONAL_HOME%\actional-instrument.cmd -all -uplinkDir C:\tmp -debug -log c:\tmp\instrument.log
```

You will see a great deal of information in this file, but the key data you may look for is highlighted in this sample log

```
2010-09-09 09:11:43.468-0600 INFO Using PlugMaker version 8.1040.29.20
2010-09-09 09:11:43.468-0600 DEBUG PlugMaker garbage collector executing every 60000ms.
2010-09-09 09:11:43.468-0600 DEBUG Instrumentation will be done with these AOPs:
    com/actional/lg/interceptor/sdk/helpers/InterHelpJaxJIT.aop
    com/actional/lg/interceptor/axis/Axis-jit.aop
    com/actional/lg/interceptor/http/client/jakarta2.aop
    com/actional/lg/interceptor/jdbc/jdbc.aop
    com/actional/lg/interceptor/jms/jms.aop
    com/actional/lg/interceptor/tomcat/tomcat.aop
    sun/rmi/server/rmi.aop
    com/actional/lg/interceptor/j2ee/j2ee.aop
2010-09-09 09:11:43.468-0600 DEBUG Current directory: C:\apache-tomcat-6.0.16\bin\
System properties:
    awt.toolkit=sun.awt.windows.WToolkit
    catalina.base=C:\apache-tomcat-6.0.16
    catalina.home=C:\apache-tomcat-6.0.16
    com.actional.aop.deployjars=actional-jdbc-interceptor.jar,actional-jms-interceptor.jar,actional-axis-interceptor.jar
    com.actional.aop.jarBase=C:/Insight/InsightAgent/interceptors/tomcat/
    com.actional.aops=com/actional/lg/interceptor/sdk/helpers/InterHelpJaxJIT.aop,com/actional/lg/interceptor/axis/Axis-jit.aop,com/actional/lg/interceptor/http/client/jakarta2.aop,com/actional/lg/interceptor/jdbc/jdbc.aop,com/actional/lg/interceptor/jms/jms.aop,com/actional/lg/interceptor/tomcat/tomcat.aop,sun/rmi/server/rmi.aop,com/actional/lg/interceptor/j2ee/j2ee.aop
    com.actional.lg.interceptor.config=C:\tmp
    com.actional.lg.interceptor.debug=true
    com.actional.lg.interceptor.sdk.PlatformType=600
    com.actional.log.file=c:\tmp\instrument.log
    file.encoding=Cp1252
    file.encoding.pkg=sun.io
    file.separator=\
```

First of all, you notice that there is a "PlugMaker" message at the very top. This indicates that the instrumentation is correctly installed in the container. The second highlighted line shows that the uplink.cfg parameter is set and we can validate that this is the directory that the Analyzer is also using.

If this information is verified, you may need to save this log file as well as the "support.zip" file (discussed in later section) to send to Software AG Global Support.

Note: do not forget to remove the -debug flag after you have resolved this issue

3 - Audit the Agent events

What this will do is put in the event log of the analyzer all the agent events sent by the interceptors. If no agent events show up, then something is really wrong, so go to step four, "Is plugmaker properly hooked in?". If you have some agent events, try to identify if you see agent events for the interceptors you are expecting. You can get a clue by looking at the URL of the agent event.

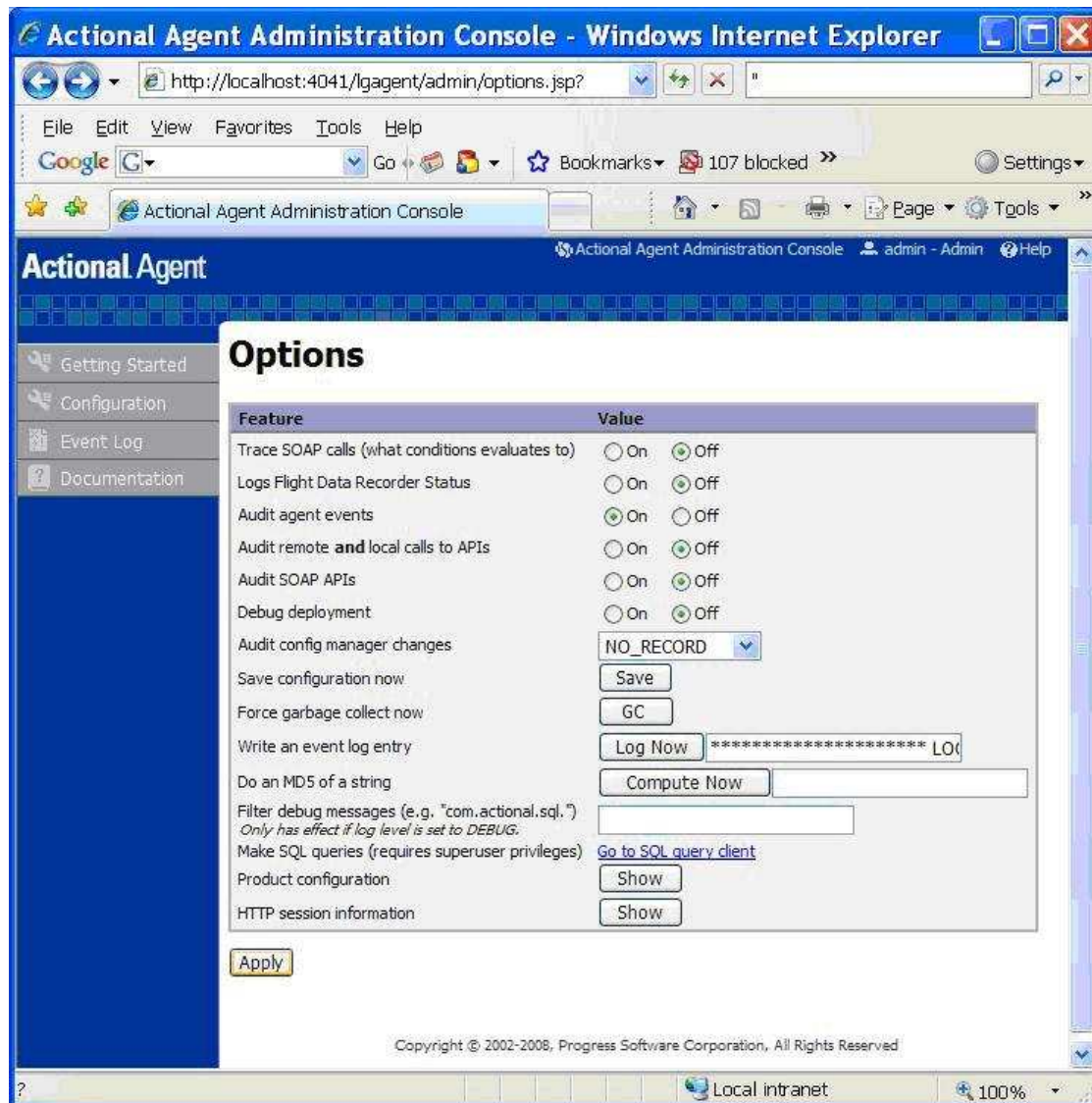
Looking at an agent event log message you can learn a few details:

- Analyze which Interceptor Events are being reported. This helps to confirm if the interceptor is reporting the correct traffic.
- See if the message body is reported for auditing.
- See which message field values are reported.
- If the Interceptor SDK is in debug mode, you should see the details of when the event was created here. You will see a call stack trace that gives you hints about which portion of the interceptor kicked in.

Turning on agent event auditing is done through a hidden page in the Actional Agent UI. When you login to the agent through a web browser, you use a URL such as this:

`http://localhost:4041/lagent/admin/options.jsp`

Where localhost is the name of the system and 4041 is the port you designated for the agent during creation of the profile. You will see a page similar to this:



Set the **Audit agent events** radio button to **On** and click **Apply**. Note that this setting is not persistent so if you restart the agent, you will need to do this again.

Results

Once you have done that, after creating traffic through the test scenarios, on the **Event Log** page under the **Report** tab, after a few seconds you will see agent events like those shown below in the log:

Actional Agent Administration Console - Event Log - Report - Windows Inte...

http://localhost:4041/lagent/admin/logging/eventlogs.do

File Edit View Favorites Tools Help

Google G Go Bookmarks 107 blocked Check Settings

Actional A... Internet Expl... Actional Serv...

Actional Agent

Report Configure Logging

Link to this page

Event Logs

Basic Search
37 entries in the last hour.

All logs Last Week Last Day **Last Hour** Clear Filter

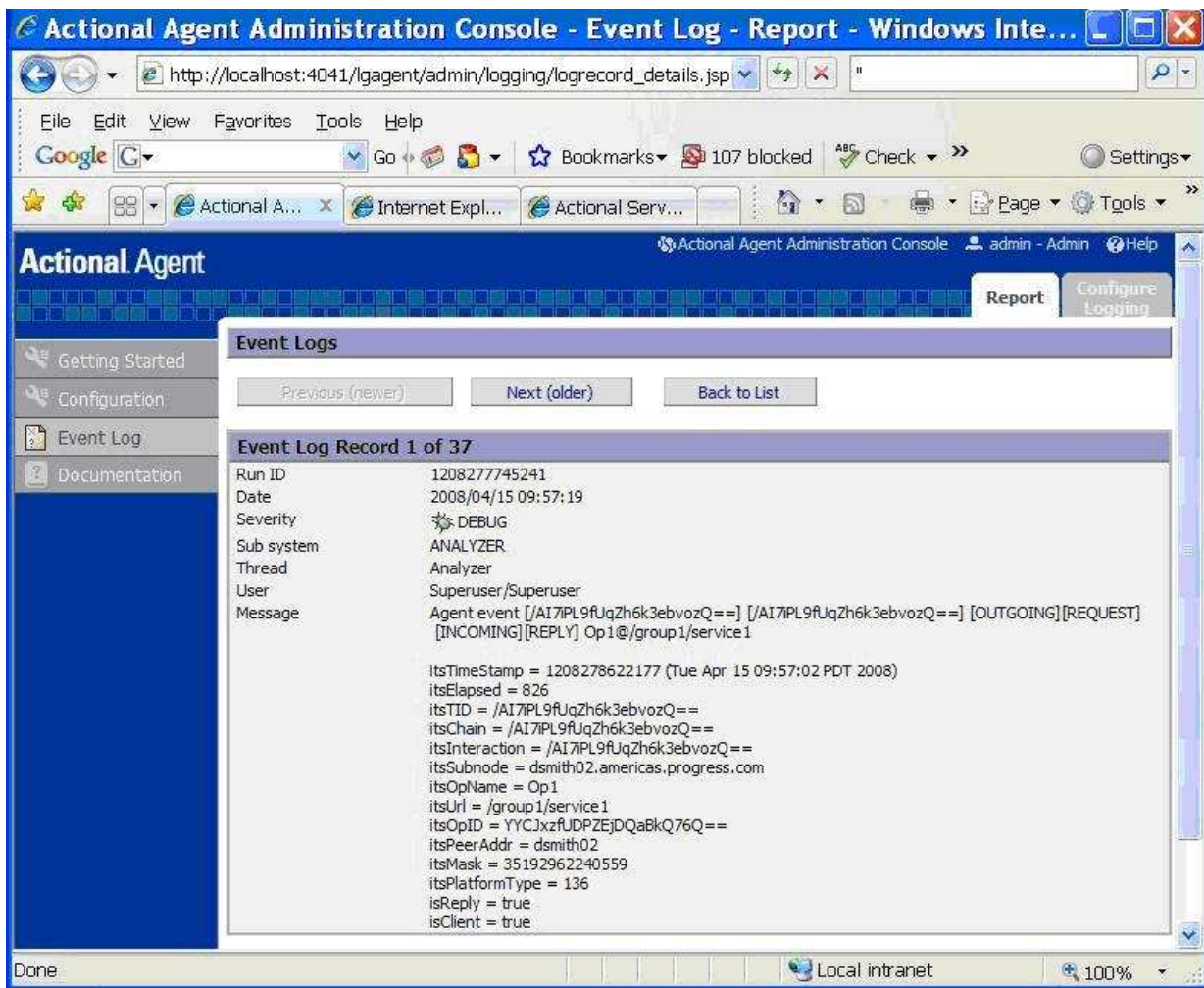
Export Fields... Refresh

50 per page

Severity	Date	Sub system	Message
DEBUG	2008/04/15 09:57:19	ANALYZER	Agent event [/AI7PL9fUqZh6k3ebvozQ==] [OUTGOING][REQUEST] [INCOMING] [REPLY] Op1@/group1/service1
DEBUG	2008/04/15 09:57:19	ANALYZER	Agent event [/AI7PL9fUqZh6k3ebvozQ==] [OUTGOING][REPLY]
DEBUG	2008/04/15 09:57:19	ANALYZER	Agent event [/AI7PL9fUqZh6k3ebvozQ==] [INCOMING][REQUEST] [INCOMING][REPLY] Op2@/group2/service2
DEBUG	2008/04/15 09:57:19	ANALYZER	Agent event [/AI7PL9fUqZh6k3ebvozQ==] [INCOMING][REQUEST] [OUTGOING][REPLY] Op2@/group2/service2
DEBUG	2008/04/15 09:57:18	ANALYZER	Agent event [/AI7PL9fUqZh6k3ebvozQ==] [INCOMING][REQUEST] Op1@/group1/service1
DEBUG	2008/04/15 09:56:30	ANALYZER	Request for dimensional data returning 8 bytes
DEBUG	2008/04/15 09:56:30	ANALYZER	Request for historical data returning 1218 bytes
DEBUG	2008/04/15 09:56:30	ANALYZER	Request for historical data returning 1218 bytes
DEBUG	2008/04/15 09:55:30	ANALYZER	Request for dimensional data returning 8 bytes

Local intranet 100%

You can drill down to see details by clicking on one of the lines:



You can also examine the results by looking in the Agent log itself. The log file has this location and file name format:

`<agent_install_directory>/logs/ActionalAgent_<date><timestamp><number>.log`

Example: ActionalAgent_20080414_110229_000.log

Here is the same agent event from the Actional Agent log:

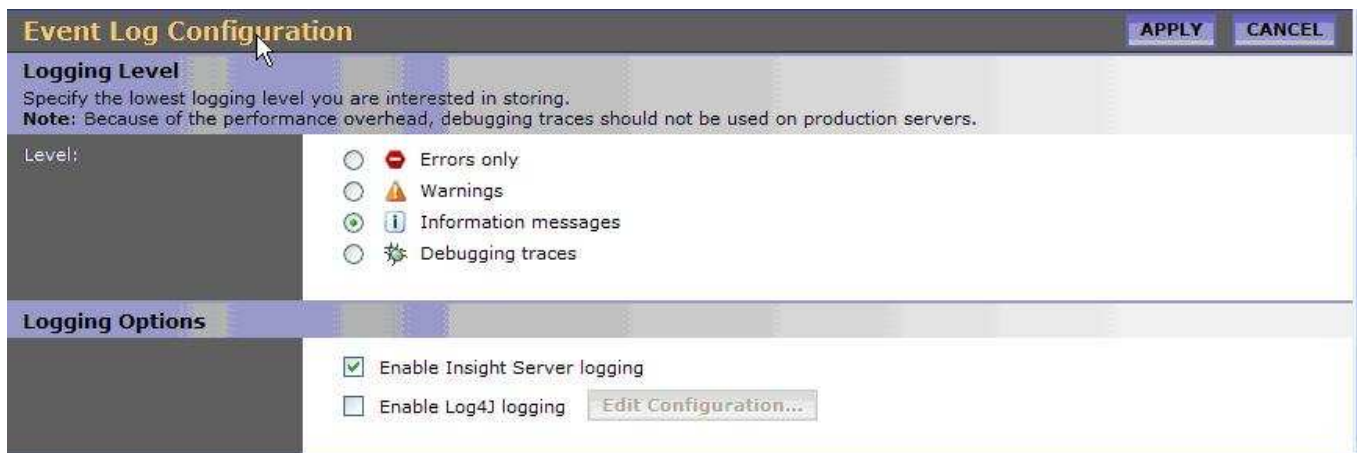
```
<event>
<date>2008/04/15
09:57:19</date><severity>DEBUG</severity><reqid></reqid><thread>Analyzer</thread><user>Superuser/S
uperuser</user><runid>1208277745241</runid><subsys>ANALYZER</subsys>
<msg>Agent event [] [] [OUTGOING][REQUEST] [INCOMING][REPLY] Op1@/group1/service1

itsTimeStamp = 1208278622177 (Tue Apr 15 09:57:02 PDT 2008)
itsElapsed = 826
itsTID = /AI7iPL9fUqZh6k3ebvozQ==
itsChain = /AI7iPL9fUqZh6k3ebvozQ==
itsInteraction = /AI7iPL9fUqZh6k3ebvozQ==
itsSubnode = dsmith02.americas.progress.com
itsOpName = Op1
itsUrl = /group1/service1
itsOpID = YJCJxzfpUDPZEjDQaBkQ76Q==
itsPeerAddr = dsmith02
itsMask = 35192962240559
itsPlatformType = 136
isReply = true
isClient = true</msg>
</event>
```

If the events are reported, but nothing is appearing in Insight Server, then try the following.

Events Being Lost?

In Insight Server, change the logging to debug level and wait for a gather. This is done by opening the Insight Server Console and navigating to Configure->Platform->Event Log.



Event Log Configuration [APPLY] [CANCEL]

Logging Level
Specify the lowest logging level you are interested in storing.
Note: Because of the performance overhead, debugging traces should not be used on production servers.

Level:

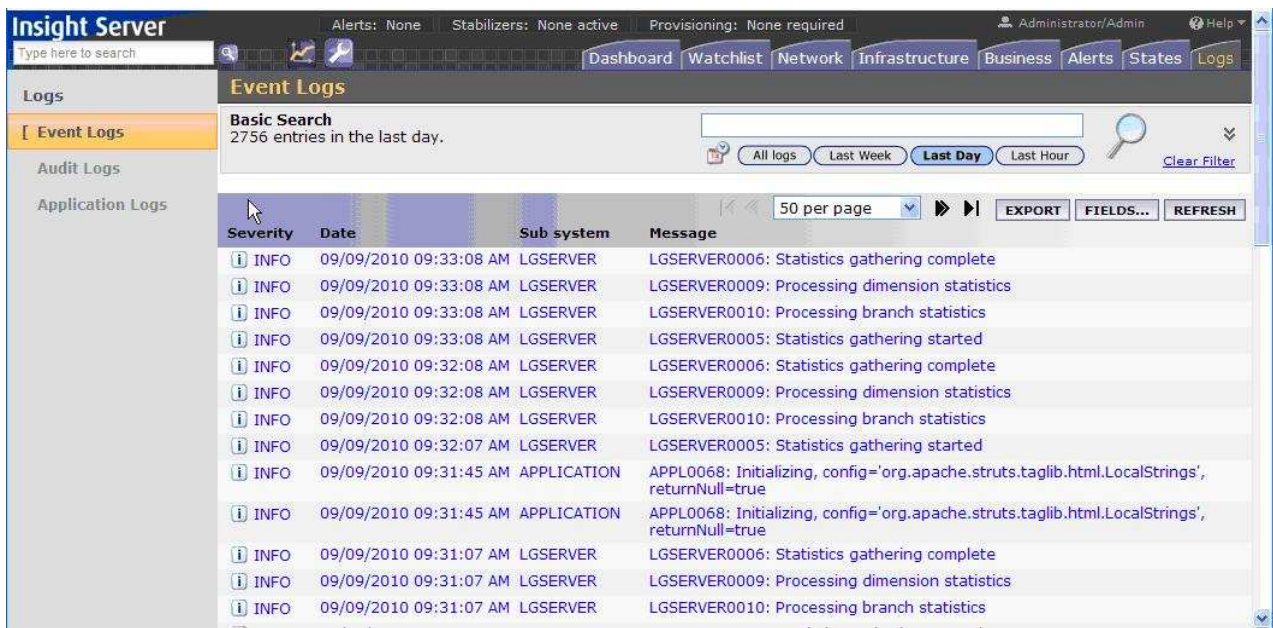
- ☐ Errors only
- ☐ Warnings
- ☒ Information messages
- ☐ Debugging traces

Logging Options

- ☒ Enable Insight Server logging
- ☐ Enable Log4J logging

[Edit Configuration...]

Click on "Debugging traces" and **APPLY**. Next navigate to the log page via Manage->Logs->Event Logs.



Do you see log messages about dropped events? If so, then that could hint at a configuration or interceptor issue.

4 - Is PlugMaker properly hooked in?

Upon startup, PlugMaker printed the following informational message on stderr:

```
Using PlugMaker version a.bbbb.cc.ddd
```

If you don't see this message, then PlugMaker was not properly deployed. If the procedure was to modify scripts, scrutinize the changes you've made. Things to look for:

- case sensitivity on Linux/Unix for directory names and file names.
- that the paths are correct.
- sourcing a script is different in SH (.) then in CSH (source).
- that the account used to execute the application server has permissions to access our scripts and JARs.
- **A typical mistake made on Linux/Unix** is to invoke the function `actional_instrument` on the same line as sourcing the script. For example:

INCORRECT:

```
. /opt/agent/actional-instrument.sh actional_instrument -all
```

CORRECT:

```
. /opt/agent/actional-instrument.sh
actional_instrument -all
```

Sourcing `actional-instrument.sh` defines a function called `actional_instrument` in the shell environment. On the second line, you invoke that newly defined function with the parameters that describe the

interceptors that you want to turn on. The function will then define the necessary environment variables that will hook in PlugMaker in the JVM and thus turn on instrumentation.

- if you have moved the agent's installation around, there is a variable called `LG_HOME` that will most likely be out of date. Here is an excerpt from the script:

```
if [ "$ACTIONAL_HOME" = "" ]; then
  LG_HOME=USER_INSTALL_DIR
  ACTIONAL_HOME=${LG_HOME}/interceptors/jboss
fi
```

`LG_HOME` is set at installation time by the installer to point at the agent's installation directory. The installer replaces `USER_INSTALL_DIR` with the agent's location. If it gets outdated, then define `ACTIONAL_HOME` to point at the directory where the interceptor jar files are found. If you have moved the agent's installation directory to `/opt/agent`:

```
ACTIONAL_HOME=/opt/agent/interceptors/jboss
. $ACTIONAL_HOME/actional-instrument.sh
actional_instrument -all
```

5 - look for stack traces

See if the Insight classes appear in any of these call stacks. If `"com.actional.plugmaker.*"` shows up, then instrumentation probably went wrong.

If `"com.actional.lg.interceptor.*"` shows up, then there is probably an interceptor that is misbehaving (most likely due to an unsupported version of the platform).

6 - Make sure that AOP files get parsed

With the `-debug` flag set (discussed in earlier section), you can search for the word "Parsing" in the log file to see if there are any problems parsing the AOP files.

Make sure that your AOP files are found and get parsed by searching for "Parsing".

7 - Generate a support.zip file

If it is not working yet, turn on debug mode, auditing and debugging capture (as described above), generate some activity through the instrumented container(s) and then generate a "support.zip" file.

The support.zip file is generated by accessing the URL:

```
http://localhost:4040/lgserver/admin/support.jsp
```

You can filter the support.zip file, but in most cases, you will just click the **Generate Support.zip** button at the bottom. Save the zip file into a directory.

You should include any container log files you have generated (or console output if you do not use -log).

This file will be requested by Software AG Global Support when you report a support incident.

To avoid a lot of back and forth, it is useful to describe in as many details as possible what it is that you are expecting to see as far as traffic goes (what are the nodes, what do you expect to see under these nodes, what calls you expect to see between these nodes). Screenshots are often very helpful.

Though it is clearly understood that is not always possible, having fully functional unit tests will greatly reduce the time required to provide you with a fix or at the very least acknowledge the problem and provide a workaround. If not, another possibility is to provide code excerpts and such.

If you customized the scripts that are bundled with the various applications, describe the customizations you have done and the process around these customizations (how the various scripts are invoked and such).

8 - Remove all flags you inserted

When you've gathered enough diagnostic information, remove all debugging tools you may have turned on. These will most likely have a performance impact on the applications.

Namely:

- Do not audit agent events;
- Put back the event log level to its original state;
- Turn off debug mode of the interceptors;
- Don't capture the instrumented classes;
- Turn off debug mode of the interceptors.

Other products - Deployer

Common Problems

Error connecting to Integration Server: localhost:wm.deployer.resource.gui.IS:ping

If you see this error when testing a connection (or attempting to deploy) to an Integration Server or Process Model server, ensure that the WmDeployerResource package is installed on the target server.

Missing shared code after deploying Java services

For Integration Server deployments, shared code or content for Java services is sometimes not deployed. This happens if the package was deployed in "partial mode". In a situation where the shared code within a package may be changing, it is suggested to either deploy the full package or to explicitly include the necessary files (.class, node.ndf, etc.).

Question mark (?) appears when performing a dependency check in a process model deployment set

This typically means that a logical server being used in the process model is either not configured in the WmDesigner package or is unreachable.

Assets in My webMethods Server are not shown in the deployment set

If some expected assets are missing when a My webMethods Server deployment set is created, set the My webMethods Server cache timeout to 0 in the project settings to disable the cache.

Process model is not operational after deploying into the target server cluster

Deployer 7.1.2 Fix 1 and earlier contains a setting to indicate an "automatic" or "manual" deployment to an Integration Server cluster. Even with this setting set to "automatic", the model would not be operational on all target nodes under certain conditions. The quick solution is to call the pub.prt.admin:scanPackage service in the WmPRT package and pass the package name associated with the process model. If you are using Deployer 7.1.2, the longer-term fix is to apply Deployer 7.1.2 Fix 2 (DEP_7-1-2_Fix2). This fix significantly changes and simplifies the process of deploying to a cluster.

General Recommendations

Things to Check before Deploying a Process Model

- For all of the logical servers that were used when creating the process model with Designer, ensure that these logical servers are configured in the WmDesigner package. The logical server names should be the same on the target as well as on the source.
- If your process model contains My webMethods Server tasks, ensure that the My webMethods Server settings are configured properly on the WmTaskClient package Home page on both the source and the target machines.
- If you are using My webMethods Server for process model monitoring, ensure that the My webMethods Server settings are configured properly on the WmMonitor package Home page on both the source and the target machines.
- If you are deploying an Integration Server trigger or a process model to a cluster and you have not yet applied Deployer 7.1.2 Fix 2, check whether a single Integration Server in the target cluster is designated as the "primary" and, on that node, the extended setting "watt.server.cluster.aliasList" is set. This property consists of a comma-separated list of all nodes in the cluster and allows for coordination of objects in that cluster.

Inspecting a Deployer Build

To inspect a deployer build, locate the project in the `<webMethods_install_directory>/IntegrationServer/WmDeployer/persist/projects/` directory on the Deployer server. This file is a zip file that you can rename and open with any zip utility.

Among the contents of this zip file is a series of Base-64-encoded .bin files. Decode these .bin files using Base-64 decoding to reveal their contents in XML format. This XML content may contain even more elements that are Base-64-encoded text. Decode these elements using Base-64 decoding to reveal their contents in XML format.

Observing Deployer Communication

If you experience problems which indicate that there might be communication issues among the different Integration Server nodes (source, target, and WmDeployer engine host), then please capture the SOAP traffic. The SOAP traffic can be captured from any Integration Server node by setting the "watt.server.SoapRPC.debug" property to "true" and restarting Integration Server. Note: If you are running the Deployer engine on either the source or target of the deployment then you will have only two nodes.

Project Settings Tips

- To enable a process model for analysis or execution on the target machine, set "Enable Process Model for Execution / Analysis" to "yes".
- To deploy task portlets from My webMethods Server with their access control lists, set "Export Access Control Lists" on the Deployer project to "yes".
- To prevent My webMethods Server from generating aliases automatically for all portal assets, set "Auto Generate Aliases" to "no".
- If you have created My webMethods Server pages and portlets anywhere other than the public folders and you do not see them in the asset tree in Deployer, check the My webMethods Server configuration properties. The "Root Folder Aliases" property is set to "folder.public" by default. Change this property to reflect the location of the created pages and portlets.
- Ensure that the client group on the Broker Server is the "admin" client group.
- Select an appropriate Trading Networks rule setting for the Deployer project. "Merge" means that the existing rules will be appended to existing rules on the target after deployment. "Replace All" means that existing rules will be deleted from the target prior to the actual deployment of the rules.

Information to Gather when Opening a Support Incident

- Describe what you are trying to do and what is not working.
- Describe the kind of solution are you deploying and identify all of the runtime components (My webMethods Server, Process Engine, Broker, and so on) that are involved.
- Provide versions, patch details, and log files for relevant source and target servers (Integration Server, My webMethods Server, Broker Server, and so on).
- For the Integration Server hosting the Deployer, provide versions, patch details, log files (with Deployer log facility set to TRACE if possible), and Deployer project reports.
- If possible, provide the Deployer project build. Software AG Global Support can potentially import the build and attempt to reproduce the issue in-house.

In addition to this information, a Deployer-focused troubleshooting guide is available on Empower (at <https://empower.softwareag.com/KnowledgeCenter/TechnicalPapers/default.asp>).

Other products - Developer

Debugging the Developer Client

Just as with Integration Server, you can use the startup parameters `-debug` and `-log` to run the Developer client in a debug mode.

For example:

```
developer.bat -debug 6 -log none
```

or

```
developer.bat -debug 9 -log C:\temp\Developer_debugLog.txt
```

Note that:

- Debug verbosity changes at levels 1, 4, 6, and 9
- It is rare that you would need to see the HTTP headers for the traffic between the Developer and Integration Server, but they are available if necessary.

Common Problems / Typical Causes

Problems while Looping in flow

Looping over record lists can result in errors as a result of misunderstandings in how the input and output lists are handled in a loop.

The most important thing to remember is that the input and output lists are instantiated one dimension smaller inside the loop. For example, if you specify a record list in the **In Array** of the loop, you will be dealing with a record inside the loop. Similarly, if you specify a String list, the loop will only contain a String – specifically, the *n*th String in the list on iteration *n*. If you specify a record table (a list of record lists) in the **In Array** of the loop, you will be referencing a record list inside the loop.

This is important because if you have a record list in the **In Array** and you see a record list while inside of the loop, something is wrong and needs to be corrected.

Another common looping problem is trying to map data directly into or out of list objects. This can successfully be accomplished as long as you do one of two things:

- Specify the index in the map.
- Loop over the list.

Looping over a list to map only specific elements from the In Array to the Out Array, however, will have the following undesired results:

- An Out Array with the same number of elements as the In Array
- Specific elements mapped
- Remaining elements set to null

The correct way to do this would be to leave Out Array blank and instead use `pub.list:appendToStringList` and `pub.list:appendToRecordList` to append the specific elements to a new list.

Error Handling Using Flow Sequences

To handle errors in flow in a try-catch manner, three sequence operations are used:

```
SEQUENCE1 (main sequence, set to exit on SUCCESS)
SEQUENCE2 (logic sequence, set to exit on FAILURE)

Service1
Service2

?
SEQUENCE3 (error sequence, set to exit on DONE or as required)
pub.flow:getLastError
EXIT "$flow" and signal FAILURE
```

Pipeline Management

The following are techniques you can use to debug errors that occur when a service executes (for example, the service does not produce the expected results or it throws an exception). You can use these techniques individually or in combination.

Stepwise Execution

If the error occurs in a flow service, use the step commands in Developer (that is, **Test > Step** and **Test > Step Into**) to execute the service one step at a time. By examining the Results tab after each step, you can quickly pinpoint where the data in the pipeline begins to diverge from the expected results.

Viewing the Results Tab

Keep in mind that the pipeline you see in Developer may not be an exact representation of the contents of the pipeline at run time. For example, if you design a service that maps `variableA` to `variableB`, you will see those variables in Developer. However, if the underlying code does not actually produce `variableA` at run time, the runtime pipeline for that service will not match the representation you see in Developer. Similarly, the underlying code may add variables to the pipeline that do not appear in Developer. To determine the exact contents of the pipeline at run time, execute the service in Developer and view the Results tab.

Saving the Pipeline

Use the `pub.flow:savePipeline` or `pub.flow:savePipelineToFile` services to capture the contents of the pipeline at run time. During debugging, you can use the `pub.flow:restorePipeline` or `pub.flow:restorePipelineFromFile` services to restore the pipeline to that state. This technique is especially useful when, for debugging purposes, you need a copy of the data that a partner is posting to your server. Warning: since the aforementioned services will write to memory or file, non-serializable objects will not be persisted across a save/restore. Be sure to disable such steps in your service before putting it into production. Starting in 8.0.x, pipeline save/restore can be toggled as a service property in Developer (rather than having to enable and disable steps in the code).

Tracing the Pipeline

The service `pub.flow:tracePipeline` takes a snapshot of the pipeline (variable names, types, and contents) and writes it to the server log. This is particularly useful when dealing with more complex or custom java object types which can not be differentiated in the Developer tool (for example, if the icons are the same).

Debug Log

If you are debugging a flow service, insert the `pub.flow:debugLog` service at strategic points in the service. The `debugLog` service writes messages in the server log. By examining the messages posted to the server log, you can tell you whether a particular code path was executed. This technique is especially helpful when debugging a flow service that has deeply nested services. Remember to remove or disable any `debugLog` steps from your service before putting it into production.

Writing to Console

If you are debugging a Java service, call the `System.out.println` method at strategic points in your code. This method writes a message to the console when it executes. By watching the messages that `System.out.println` generates, you can determine whether a particular code path is executed. Remember to remove the calls to `System.out.println` method from your service before putting it into production.

Information to Gather When Opening a Support Incident

For a Developer-related issue, please provide the following information:

- A copy of Developer > Help > Support Information (this lists the version and fix levels for the Developer client as well as any connected Integration Servers)
- The archived package(s) and any required packages (if providing a reproducible package, try to simplify the scenario or service so that it can be run "stand-alone". The simpler the provided service, the quicker it can be processed by Software AG Global Support)
- Server-side logs and configuration files (these can be easily packaged using `wm.server.admin:getDiagnosticData` utility service)
- If the Developer tool hangs, a [java thread dump](#) generated for analysis

Other products – Software AG Installer

Obtaining Diagnostics

Software AG Installer Version

To view the Software AG Installer version:

- Start the Software AG Installer.
- Click the Advanced Options button on the first page.
- Click the About tab.

or type the following on the command line:

```
java -jar SoftwareAGInstaller80SP2.jar -version
```

These will reveal a string like: "Software AG Installer Version 8.0.2.0.113 110909".

Installer Debug Logs

To generate a debug log from the Installer client, run the installer and go to Advanced options > Logging.

Set the Logging Level to Verbose to capture detailed output.

This output is not easily read by users, but will be of use to Software AG Global Support in extreme installer issues.

Test Connectivity to Installer Servers

If you suspect any connectivity issues during installation, you can try to connect to the installer server directly and perform the following tests to validate connectivity:

- ping sdc.softwareag.com and sdc-hq.softwareag.com
- telnet sdc.softwareag.com 80 and telnet sdc-hq.softwareag.com 80
- Open a browser to one of the following, depending on whether you want to install 7.x or 8.x products:
 - <http://sdcsdc.softwareag.com/cgi-bin/dataservewebM71.cgi>
 - <http://sdcsdc.softwareag.com/cgi-bin/dataservewebM802.cgi>
 - <http://sdcsdc.softwareag.com/cgi-bin/dataservewebM82.cgi>

The expected response the browser would receive is a blank page, showing "ERROR: 1".

Thread Dump of the Installer

The installer is a Java application. In rare circumstances, such as when the installer appears to be hung, it might be useful to obtain a Java thread dump. The steps for obtaining a Java thread dump might vary depending on the platform. See the Advanced Troubleshooting Guide chapter on Integration Server hang for instructions on how to get a thread dump on a hung Integration Server Java process. The steps will be the same.

Common Problems

Some Products Not Listed

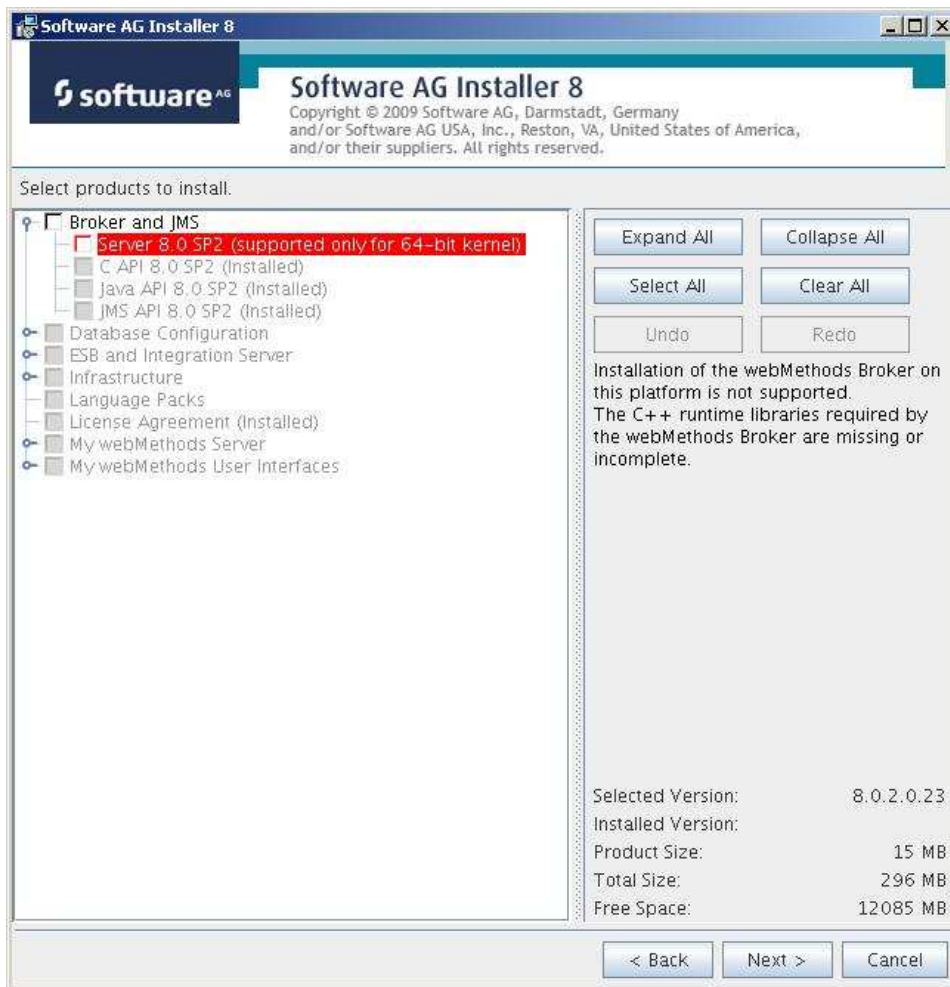
If expected products do not appear in the product list of the installer after your user name and password have been authenticated, a few different things might be happening:

- You are not licensed for those products.
- You are licensed, but your account has not been updated properly.
- The products are listed in a different area of the product tree (for example, because they have changed names between versions).

Open a Support Incident with Software AG Global Support to clarify.

Cannot select Broker for installation on 32-bit Linux

Installer shows the following error message when trying to select the "Broker" component during installation:



"Installation of the webMethods Broker on this platform is not supported. The C++ runtime libraries required for the broker are missing or incomplete."

- Check that "compat-libstdc++" is correctly installed.

Installing From a .jar File

When installing using the *.jar installer client, make sure to use a supported JRE. The supported JRE versions for use with the Installer client are listed in the Using the Software AG Installer guide on <http://documentation.softwareag.com>.

Proxy Problems

If you are running the installer from a machine that sits behind an outbound proxy, you might need to specify proxy server information in the installer. A good indicator is to look at proxy settings of your browser on the target machine.

Corrective Actions

Use the Latest Installer Version

Many installation problems can be avoided by using the latest version of the installer client. It is typically a good idea to obtain the latest version from the Empower Software Download Center

<https://empower.softwareag.com/Products/DownloadProducts/default.asp> which is updated periodically.

Information to Provide When Opening a Support Incident

Please provide the following information when submitting an Installer-related Support Incident to Software AG Global Support.

- Full version of the installer client being used (see above).
- Operating system the installer is running on.
- User name under which the installer is running.
- Exact, full error message (a screenshot might be sufficient), with stack trace if there is one.
- Other relevant details (for example, whether you are using a proxy, building an install image, running in console mode, running in script mode).
- Verbose debug log file that was generated (see above).

Other products - MDM OneData

Common Problems

Admin Workbench

OneData Installation

OneData Welcome email: The URL link generated in the Welcome email is not the correct one

If OneData link in the Welcome email is not correct, follow the steps to correct it:

1. Locate the 'onedata.properties' file by navigating to onedata/webapp/web-inf/config directory
2. Edit the file and set the host path for parameter 'onedata.application.uri' as `http://hostname/onedata/`

JDBC driver: Find the JDBC Driver used in OneData

OneData uses the JDBC Driver version provided by the application server. The JDBC driver file obdbc5.jar can be found in onedata/webapp/web-inf/lib directory.

- If you are using WebLogic Application Server, then take a look at the file META-INF\MANIFEST.MF inside %BEA_HOME%/weblogic81/server/lib/ojdbc5.jar
- For WebSphere, the file is found in location ORM_HOME/lib.
- For Tomcat 5.5 this file is found in <TOMCAT_ROOT>/common/lib and for Tomcat 6.0 it is found in <TOMCAT_ROOT>/lib.
- For JBoss application server, this file is in the directory %JBoss_HOME%/lib/endorsed.

Upgrading to a new version of OneData

We have just upgraded to OneData 8.X from 6.X. We did this by downloading all versions between 6.X and 8.X, running the scripts, but then just deployed the 8.X war file in application server. Could you please confirm that this is a valid sequence? Also, is there any restriction on the number of versions our installation can be behind the current version under the maintenance agreement.

Yes, the sequence of the upgrade steps you've done is correct. This is true for most upgrades - the upgrade SQL scripts from all in-between releases should be applied, but only the latest application should be deployed.

And you can upgrade to any newer version of OneData without any restriction.

OneData Login

Login Exception: javax.security.auth.login.LoginException: No LoginModules configured for OneData

If you are getting No LoginModules configured exception, then the problem may be because of a missing or incomplete JAAS configuration, or the JDK file syntax.

To verify this, go to `onedata/webapp/web-inf/config/login.cfg`. Check if `OneDataLoginModule` along with the flag is specified in OneData application block inside `login.cfg`. An example is given below:

```
onedata {
com.datafoundations.onedata.security.auth.module.OnedataLoginModule required;
};
```

Security Exception: java.lang.SecurityException: Unable to locate a login configuration

This problem may be because of a missing or incomplete JAAS configuration, or the JDK file syntax.

To resolve this problem, go to `$JAVA_HOME/jre/lib`. Edit the `java.security` file. Verify that the `login.cfg` file is correctly pointing in the mapping '`login.config.url`'.

An example for this is: `login.config.url.1=file:/opt/onedata/webapp/WEB-INF/config/login.cfg`

Login Exception: javax.security.auth.login.LoginException: Missing users.properties file

To resolve this issue, check the login module configuration.

Either `login.cfg` should be used for configuring application server or application policy should be added to login-config of application server with name `OneData.<REPOSITORY_ID>`.

Login error: Unable to login. Please retry or contact your administrator

Check all connections strings and data sources.

For direct connections, check `CNCTN_STR`, `USR`, `PWD` inside `/opt/onedata/webapp/web-inf/config/repository.xml`.

For connection pool, data source should be defined in application server and JNDI name should be used for `CNCTN_STR`.

Start-up error: Application startup log shows "webMethods OneData couldn't be started."

Check `/opt/onedata/webapp/web-inf/config/` folder. The following files should be present in the config folder:

```
# Cache_clustered.ccf
# cache_standalone.ccf
# log4j.properties
# onedata.properties
# quartz.properties
# repository.xml
```

Also make sure that all files are up-to-date.

Access denied error: java.security.AccessControlException: access denied

This error could be possibly a permissions issue in the policy files. Either the `client.policy` for the application client or the `server.policy` for server side components does not have permission to set the property.

To solve this, add the permission in `client.policy` (for the application client), or in `server.policy` (for application server) for the application that needs to set the property.

By default, applications only have "read" permission for properties.

Metadata error message upon login

When I try to login to OneData, I receive an error message: Current meta data version XXX is not compatible with current code version YYY.

This could be a metadata compatibility error with code version. Please contact Software AG Global Support via eService to request that the Metadata be synchronized with the Code version.

Object Definition

Object Creation

Error when executing DDL statement in OneData

When I try to run a DDL Statement in OneData, I get the following error message: "There was a problem in executing 1 of 1 DDL statements. The following statements failed :
ava.sql.SQLException: ORA-00922: missing or invalid option".

This error usually occurs if the semicolon is missing after the DDL statement. Please check the statement and add necessary punctuations.

Automatically getting logged out when accessing an object in Data Manager

When I am accessing an object through Data Manager, I am getting logged out from the system.

This usually happens when the object has two foreign key constraints defined on the same column. To resolve this issue, configure the object such that the column has only one foreign key constraint defined.

Remote Objects - Database Sequences & Triggers

When defining a Remote Object, the database sequences and triggers are not brought over. Tried bringing the sequence ahead of the table manually, but it still didn't work. The Database Sequence column data type is defined as Numeric in the new Remote Object.

OneData currently does not have the capability to recognize sequences or triggers for remote objects. Once the Data Object has been created in the system there are two possible workarounds to enable the sequences:

1. Use the Edit Column screen in Object Definition to create a new sequence and trigger.
2. Manually create the sequence and trigger in the back-end, change the data type of the column to database sequence on the Edit Column Screen, and mark the sequence and triggers as existing.

Column name appears twice in the Filter on Nova/Nova-Echo objects

Is there a way to control the behavior of the Filter on Nova/Nova-Echo objects? I have an object with four FK constraints. On three of the columns associated with these FKs, on the filter screen, the column appears twice, once with the caption that I have entered and once with the name of the description column named in the FK constraint.

In Nova/Nova-Echo mode, for FK constraints, we allow multiple related description columns as against single description column in Default mode. As such, in Nova/Nova-Echo, these FK constraints can be further configurable.

To control the behavior of FK columns on the filter screen for Nova/Nova-Echo objects

1. Edit the FK constraint.
2. Select the popup reference icon next to Related Column Description attribute
3. Select Additional Properties in the popup screen
4. Set values for Show in Filter? attribute as per requirements and Save.

Manage Data

Data Manager

Conceptual Object - Tree View; "No description"

I have defined a Conceptual Object and configured it as a Tree View. But when I access this object in Data Manager, on the left grid, it says "No description". What am I missing?

When a Conceptual Tree is configured to be displayed as a Tree, then one column in every object which is a part of this conceptual object has to be tagged as a description column by setting the attribute "Is description column?" at the column-level.

Conceptual Object with Recursive Structures

I have created a Network Recursive Conceptual Object and configured it to be displayed in Advanced Parent-Child view. It doesn't seem to display correctly in Data Manager. Is this a bug?

If a Conceptual Object holds a Self Recursive or a Network Recursive structure, it has to be configured as a Tree view in Default mode to display accurately. In addition, the root (top-level) object in the Conceptual Object should hold the Self or Network Recursive relationships.

Blank report when trying Exporting to Excel in Data Manager

In Data Manager when I select Delivery Options -> Export to Excel or Delivery Options -> Summary report, I get a blank report.

To export records to Excel, simply edit the column definition and set the attribute 'The Show in Report?' to be displayed in the report.

In Nova mode, Export to Excel is not able to export data in .xls. But, in Default mode, this feature is working

If Nova object has a foreign key relation and when I do Export to Excel, records are not exported to Excel.

The reason for this problem could be an additional settings missing. To resolve this, please follow the instructions below:

1. Please go to the object definition. Click the 'Structure' tab. Edit the column you want to export to Excel. Select the field 'Show in Report' and select any value from Show in Summary/Show in Detail/Show in Both. You can edit all the columns in a single screen by 'Edit Multiple Columns' sub-menu.

2. Next, go to sub-menu 'Edit Constraints'. For the Foreign Key Constraint relation ship, click Edit. Select the icon next to the 'Related Description Column'. In the window that shows up, select 'Additional Properties' at the top right. Select from Show in Summary/Show in Detail/Show in Both options under the 'Show in Report' field and save this.

3. Please try Delivery Options -> Export to Excel once again.

This should resolve the issue.

Application Cache extraction function has caused an exception, please contact your administrator

While accessing an OneData object using the menu Manage -> Manage Data, I'm getting this error. How can I resolve this problem?

The issue is due to lack of allocated memory to Oracle buffers (larg_pool and shared_pool). Please increase these buffers to resolve this issue.

Data Interchange

Data Import

Data Imports - Data Imports taking hours to complete

Importing data into OneData objects in Data Manager is painstakingly slow; it takes hours for a few thousand lines to import.

Please check if System Properties -> General -> Work Area-to-Release Area Mode is set to Nova mode.

Users can now speed up data import by using Interchange Mapping profile to import data. Further speed-up is obtained by enabling 'Bypass Audit' in Processing Options and selecting 'Data Compare Criteria' as case insensitive in IM profile. Additional improvement can be obtained by bypassing other validations such as temporal validation, regular expression and rules engine, workflow, etc. If the data import is in millions, then providing larger value in 'Execution Batch Size' improves performance. Above all, since the import process is carried out in parallel, it's always advisable to use systems with multiple processors. Nova mode is not applicable to old Data Manager Import.

If the rows of import is in the order of 750K+ records, it is advisable to use "Multiple connection enabled". This will in turn enable partial commit as well.

Getting error message when trying to import XML file

When I import XML data into an object using Data Interchange >> Import from XML, the data will be imported into the entity with the corresponding table_name that was used in the XML file. If any errors are found during processing the XML file, these will be shown on the next screen.

Please try the import via Data Manager (Manage -> Manage Data -> Import -> Import from XML file) or use Interchange Mapping (Data Interchange -> Configuration -> Interchange Mapping -> XML/XSD Import file) to import XML file.

Getting error message "Failed to create file in the webMethods OneData temporary directory" while importing a file or creating an Interchange Mapping profile.

To resolve this error, please try the following:

1. Please check with Sys Admin that the user who's logged into the machine has the required privilege to create/modify/delete folders.
2. There could be chance that the max. count for temporary folders in "onedata/temp/C001" has reached. If that has reached then delete some temporary folders that are older than 3-4 days.

Deployment

Hook Definition

iHook RMI Problem

RMIIHookServer error: Listen failed on port

I tried to recapture sample from the OneData training for iHooks.

I tried to setup a RMI Server to test my iHooks as mentioned in the training. Set the OneData RMI port to 4099 in onedata.properties, but attempt to run the server, I get following error message:

```
RMIIHookServer error: Listen failed on port: 4099; nested exception is:
java.net.SocketException: Unrecognized Windows Sockets error: 0: JVM_Bind
```

To resolve this issue, please try the following changes:

1. Disable the security manager in java.security.
2. Choose another port because 4099 was already used by another application.

Information to Gather When Opening a Support Incident

Environment Details about Where the Problem is Occurring

Version of OneData application, version of application server and repository metadata version. You can get this information by clicking About webMethods OneData on the top right corner of OneData Home page. Providing additional information such as platform details, Oracle database version, Java version in DEV, TEST and PROD environment will also be useful.

Log File Associated with the Problem Component

- log4j.properties resides in the OneData CONFIG folder
- All loggers are set by default to WARN threshold level
- To enable detailed logging on a specific logger, set it to DEBUG level

Logger Name in config file	Module on which log is generated
log4j.logger.onedata.Security	Security like Users, Roles, User Role assignment, etc.
log4j.logger.onedata.DataManager	Old Data Manager
log4j.logger.onedata.ChangeManager	Change Request, Workflow
log4j.logger.onedata.ObjectDefinition	Object Definition
log4j.logger.onedata.Misc	Data Audit, LDAP, License, Administration like System Property & Connection Management, Transfer Metadata, Email notification, Real-Time S2P (Old S2P), Workflow Escalation
log4j.logger.onedata.Distribution	Distribution
log4j.logger.onedata.S2PHandler	Outbound XML, Old S2P, Nova S2P
log4j.logger.onedata.ImportManager	Old and new Import (including CDI Import), Exception Queue
log4j.logger.onedata.Global	RMI, Scheduler startup/failure Log, Personalization Cache for Eager metadata cache, Licensing Mgmt, Clear Cache for repository, Application Startup, RESTful Web Services
log4j.logger.onedata.Reports	Reports
log4j.logger.onedata.Objects	Remotely Hosted Object
log4j.logger.onedata.Hook	Hooks
log4j.logger.onedata.ForumManager	Discussion Forum

log4j.logger.onedata.Registry	Registry Import
log4j.logger.onedata.Activity	Job Scheduler Activity
log4j.logger.onedata.Notification	New Notification
log4j.logger.onedata.MatchingEngine	Matching Engine, Data Quality (Trillium, D&B) related actions
log4j.logger.onedata.NovaManager	Nova
log4j.logger.onedata.InMemoryCache	InMemory
log4j.logger.com.dfi.od.validation	New nova validation framework
log4j.logger.onedata.RemoteCache	Remote (cluster) JCS Cache

Module specific logger request

For New Import issues

log4j.logger.onedata.ImportManager

log4j.logger.onedata.NovaManager

log4j.logger.com.dfi.od.validation

For Nova DM related issues

log4j.logger.onedata.NovaManager

log4j.logger.com.dfi.od.validation

For Workflow and S2P related issues

log4j.logger.onedata.ChangeManager

log4j.logger.onedata.S2PHandler

log4j.logger.onedata.Misc

For Deployment related issues

log4j.logger.onedata.Reports

log4j.logger.onedata.Distribution

Platform-wide - Disclaimer

Links to any sites or documentation, external to Software AG, are provided for informational purposes only. Software AG does not endorse or support information available at these links, nor does it make any claims regarding the veracity of this information. If you encounter any problems that result from the use of information contained in a third-party vendor site, please contact that vendor.

As with any troubleshooting and diagnostic step, please test all items before implementing them in your production environment. If you are not sure about any of the information contained in this document, please contact Software AG Global Support.

Platform-wide - How to obtain a JVM Thread Dump

About Thread Dumps

A thread dump is a snapshot of the stack traces of currently running threads in the memory of a JVM process. Thread dumps can be critical in resolving issues, such as [server hangs](#) and [performance](#) issues.

To maximize the collection of diagnostic information, it is recommended that you capture thread dump information several times at a few minutes interval.

Capturing Thread Dump Information

To obtain a JVM thread dump on the respective operating system, proceed as follows:

Windows

Windows Console Thread Dumps

1. Start a command prompt.
2. As the thread dump information will not fit in the command prompt, configure the screen buffer width and height to display the complete thread dump as follows:
 - Right click the command prompt title bar and select **Properties**.
 - Click the **Layout** tab and on the **Screen Buffer Size** area, set the **Height** field to 9999 lines.
 - To avoid word wrap, set the **Width** field to 250 symbols.
 - Click **OK**.
 - In the **Apply Properties To Shortcut** window, choose **Apply properties to current window only** and then click **OK**.
3. Start the server using the command prompt.
If your server is running as a service instance, stop the service instance and re-start the server using the command prompt.
4. Make sure that the -Xrs switch does not appear in the startup command of Integration Server in the server.bat file.
5. Start the server by navigating to the \$IS_HOME/bin directory and executing the server.bat in the command prompt.
6. To obtain thread dump information, press CTRL+Break.
7. The information appears in the console window.

UNIX

- To find the process ID (PID) of an IS Java process, execute `ps -ef | grep java`.
- To obtain the thread dump, execute `kill -3 <PID>` (or `kill -QUIT <PID>`)
- If the thread dump information is not displayed in the console window, check for `nohup.out` or `javacore*.txt` files.

If you started the IS Java process in a regular command prompt, press CTRL+\ in the window in which the Java program is started.

Preparing Thread Dump Information Files

When you obtain thread dump information in the console window, you can prepare a text file which contains the result and send it to Software AG Global Support team for further assistance. To prepare a thread dump information file, proceed as follows:

- Right click the command prompt title bar and select Edit -> Select All.
- Right click the command prompt title bar and select Edit -> Copy.
- Open a text editing program, such as notepad, and paste the data.
- Save the file on your local file system.

Platform-wide - CPU

Sometimes situations occur in which an application is consuming more CPU cycles than normal. In most cases, this is just an indication that the application is busy and the issue will be resolved after the processing is completed. If the application uses more CPU cycles than expected for a long period of time, this could indicate a problem. The options for troubleshooting high CPU usage of webMethods products depend on which [Operating System] you are using, which JVM you are using, which OS tools you have installed, and which JVM options you have enabled.

General Approach

In order to find out what an application is doing while it is consuming a high level of CPU, you can check which thread(s) is/are using the CPU cycles. After taking a thread dump of the application, the stack information of the thread(s) in question will reveal what the thread was doing.

A lot of tools exist that can show information about CPU usage. Most of these tools are available within the operating system in use. However, not all tools are able to show information about an application's threads. For this troubleshooting approach it is important to have a tool that is able to display detailed information. The next sections provide information applicable to the most common operating systems.

Using tprof on IBM AIX

You can use the `tprof` command to run a trace on AIX to find the CPU usage by process and by thread (among other good information). This can be used, in parallel with thread dumps (see: [How To Obtain A JVM Thread Dump](#)), to identify what the offending thread(s) is/are doing.

To obtain a useful trace, run:

```
tprof -skex <output filename> <seconds to run profiler>
```

For example:

```
tprof -skex /tmp/outputfile 60
```

This will create a file called `outputfile.prof` containing 60 seconds worth of tracing.

The `*.prof` file has several sections. The first section shows you the CPU utilization by process:

```
Process Freq Total Kernel User Shared Other
=====
/usr/java5_64/jre/bin/java 396 96.62 0.64 0.00 85.10 10.89
wait 4 3.03 3.03 0.00 0.00 0.00
sshd: 1 0.10 0.03 0.02 0.05 0.00
lrud 2 0.09 0.09 0.00 0.00 0.00
/usr/bin/tprof 2 0.03 0.01 0.00 0.01 0.00
...
```

You can see here that the java process is taking up 96.62% of the CPU cycles. 85% of the time this process is running shared library code. The next section of the `*.prof` file shows a break down by threads:

```

Process PID TID Total Kernel User Shared Other
=====
ava5_64/jre/bin/java 786532 3534905 16.23 0.02 0.00 16.21 0.00
ava5_64/jre/bin/java 786532 2551969 16.10 0.03 0.00 16.07 0.00
ava5_64/jre/bin/java 786532 2220119 16.04 0.02 0.00 16.01 0.00
ava5_64/jre/bin/java 786532 3977349 1.76 0.01 0.00 1.71 0.03
ava5_64/jre/bin/java 786532 8171603 1.48 0.00 0.00 1.44 0.04
ava5_64/jre/bin/java 786532 6135877 0.98 0.00 0.00 0.93 0.04
ava5_64/jre/bin/java 786532 8413271 0.94 0.00 0.00 0.90 0.04
...

```

The offending threads are the first three which are taking up about 50% of the CPU time. The TID column gives you the native thread ID in decimal format. Convert this number to hexadecimal to get the Native Thread ID which is available in the Java thread dumps.

From the example above, if you take TID: 3534905, which is 0x35F039, and look that up in a set of thread dumps that was taken while the above trace was running, you see:

First thread dump:

```

"Gc Slave Thread" (TID:0x0000000118466500, sys_thread_t:0x000000011829F1A0, state:R, native
ID:0x000000000035F039) prio=5

```

Second thread dump:

```

"Gc Slave Thread" (TID:0x0000000118466500, sys_thread_t:0x000000011829F1A0, state:CW, native
ID:0x000000000035F039) prio=5

```

Third thread dump:

```

"Gc Slave Thread" (TID:0x0000000118466500, sys_thread_t:0x000000011829F1A0, state:R, native
ID:0x000000000035F039) prio=5

```

In fact, the other two threads listed under 3534905 have the same stack traces in the thread dumps. This indicates that much of the time is spent on Garbage Collection.

Using Glance on HP-UX

On HP-UX you use Glance to show a list of the currently running processes and the respective CPU usage. When you identify the process which has the highest CPU usage, browse its threads and discover which thread is responsible for the high CPU usage. The tool displays the thread ID (LWP). If you obtain a thread dump, you use this LWP ID to determine which thread in the thread dump causes the high CPU usage.

You can associate the information from Glance to the respective thread dump as it is explained in the [tprof on AIX](#) section.

Using prstat on Solaris

On Solaris you can obtain the information about the CPU usage of threads using the `prstat` command.

```
prstat -L -p <PID>
```

You can associate the information from `prstat` to the respective thread dump as it is explained in the [tprof on AIX](#) section. Note that `prstat` doesn't show thread ID's (TID), but LWP ID's; these LWP ID's match the Native ID ("nid") in the thread dump.

Using top and ps on Linux

To identify the thread using most of the CPU time, the `top` and `ps` commands can be used.

- Determine which process is using the CPU
 - With `ps`:

```
ps -ef |grep java
```

- Output of `top`:

```
Tasks: 93 total, 2 running, 91 sleeping, 0 stopped, 0 zombie
Cpu(s): 50.1%us, 0.3%sy, 0.0%ni, 49.6%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 2059752k total, 2010428k used, 49324k free, 315772k buffers
Swap: 1048568k total, 325092k used, 723476k free, 412708k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
20408 wmusr 17 0 814m 13m 8032 S 100 0.7 0:08.38 java
2561 root 15 0 10652 616 468 S 0 0.0 100:42.25 vmware-guestd
6195 wmusr 25 0 1761m 452m 23m S 0 22.5 4:32.23 java_is
20430 wmusr 15 0 12588 1056 812 R 0 0.1 0:00.01 top
1 root 15 0 10328 76 48 S 0 0.0 0:37.34 init
```

- Determine the offending thread(s) within the process
 - In the given process, list the different threads, sorted by CPU usage, and print the top 10 CPU-using threads:

```
ps -mo pcpu,pid,tid,user,args -p 20408 | sort -k1 -r | head -10
%CPU PID TID USER COMMAND
99.3 20408 - wmusr java TestThread
98.4 - 20420 wmusr -
0.7 - 20409 wmusr -
0.0 - 20429 wmusr -
0.0 - 20428 wmusr -
0.0 - 20427 wmusr -
0.0 - 20426 wmusr -
0.0 - 20425 wmusr -
0.0 - 20424 wmusr -
```

The first line shows the overall CPU usage for the process (the sum of all thread usages). The second one is for the thread that consumes most CPU (98.4%).

Note: the CPU usage is defined as the percentage of time spent running during the entire lifetime of a process. The longer the process is running, the more difficult it will be to pinpoint the thread that is currently using the CPU. In that case, using `top -H` In the third column (TID), please note the thread-id 20420 and convert this decimal value to hex (4FC4). This value maps to the "native ID" value in the Java thread dump.

- Pressing 'H' (Shift + H) in `top` while it's running will show you the thread ID's in the 'PID' column. This may be used instead of `ps`.
If the process ID is known and `top` is not running:

```
top -H -p <pid>
```

- Get the stacktraces of the threads
 - Take a thread dump of the JVM and identify the Java Thread using the hexadecimal native ID from the previous step:

```
kill -3 20420

Full thread dump Java HotSpot(TM) 64-Bit Server VM (19.1-b02 mixed mode):
...
"Sleeper_1" prio=10 tid=0x00002aaab403d000 nid=0x4fc5 waiting on
condition [0x000000004234b000]
java.lang.Thread.State: TIMED_WAITING (sleeping)
at java.lang.Thread.sleep(Native Method)
at TestThread.sleep(TestThread.java:6)
at TestThread$2.run(TestThread.java:30)
at java.lang.Thread.run(Thread.java:662)

"Worker" prio=10 tid=0x00002aaab403b000 nid=0x4fc4 runnable [0x000000004224a000]
java.lang.Thread.State: RUNNABLE
at TestThread.work(TestThread.java:16)
at TestThread$1.run(TestThread.java:26)
at java.lang.Thread.run(Thread.java:662)

"Low Memory Detector" daemon prio=10 tid=0x00002aaab4021800 nid=0x4fc2
runnable [0x0000000000000000]
java.lang.Thread.State: RUNNABLE
...
```

Now we can say that the most CPU time is used by the Java thread "Worker", in the method TestThread.work.

Using ps and dbx on UNIX

- Discover the thread ID which causes the higher CPU usage

```
ps -mp <java_pid> -o THREAD
```

- Browse for a higher CPU value (compared to the others) in the **CP** column, or use a pipe to pass the ps output to the sort command.
- [Obtain a thread dump](#). You need this information at a later stage.

```
kill -3 <java_pid>
```

- Run the dbx command.

```
dbx -a <java_pid>
```

- Locate the thread ID in dbx.

1. List all threads

```
thread
```

2. Discover the thread ID (the thread with the higher CPU usage) which you found at the beginning of the procedure (step 1).
3. Note the number in the left hand side column in the following format:.
At a later stage you use this number to obtain more information about the thread.
4. Obtain information for that thread.

```
th info $t<number>
```

Use the number from the previous step. In the result, browse for the value of pthread_t=nnn.

5. Exit dbx

```
detach
```

- Locate the discovered thread in the thread dump report.
- Convert the pthread_t=nnn value to hexadecimal.
- In the thread dump, browse for the thread identified by native ID thread: <0xnnn>. This thread causes the higher usage.

Using ProcessExplorer on Windows

The Windows Task Manager does not display information about the threads of processes. Therefore, you need an alternative tool to obtain that information. For example, you can use the ProcessExplorer tool from Sysinternals to complete this task.

- Discover the thread ID which is using the CPU.
 1. Start the ProcessExplorer tool.
 2. Select the process (java.exe).
 3. Right click the process, then select **Properties** and then select the **Threads** tab.
 4. Select the thread with the highest CPU usage value.
 5. Note the value **Thread ID** and convert it to hexadecimal.
- Obtain a thread dump.
[Take a thread dump of the process](#)
- Locate the offending thread in the thread dump.
 Search for the thread in the thread dump which has a nativeID same as the threadID (hexademical) which you discovered in one of the previous steps (step 5).

Platform-wide - Memory

Overview

Many of the webMethods products run within a Java Virtual Machine (JVM). The JVM allows you to specify the amount of memory allocated to the application by using the `-Xms` and `-Xmx` parameters. This section of the memory is the Java Runtime Heap. The application stores runtime objects, variables, and other items in the allocated Java Heap.

Another important memory segment established within the JVM is the Permanent Generation space. As per Sun, "the permanent generation is used to hold reflective data of the VM itself such as class objects and method objects. These reflective objects are allocated directly into the permanent generation, and it is sized independently from the other generations." (<http://java.sun.com/docs/hotspot/gc1.4.2/faq.html>)

The Native Heap is also an important memory segment. The JVM uses the native heap to allocate "native" memory objects. Variables/objects created by the JVM native libraries or native libraries used by the application (such as SAP JCO libraries or JDBC OCI drivers). Users cannot specify the amount of native heap to be used. The size of the native heap is approximately the difference between the addressable space and the allocated java heap (including permanent generation). In most 32-bit environments the addressable space maxes out around 2 to 3GB. With 64-bit environments however, the limit is higher enough to consider it unlimited.

As the application runs, the JVM manages memory and periodically runs a "garbage collection" (GC) to free up objects from the heap that are no longer referenced by the application.

Memory Troubleshooting typically involves the analysis of memory usage patterns or examining the contents of the heap to identify the cause of excessive memory usage or memory leaks. Heap profiling may become necessary when encountering an [OutOfMemoryError](#), excessive [garbage collection] activity, or when memory usage needs to be reduced.

The `java.lang.OutOfMemoryError` is thrown for a number of different memory-related failure events which may occur during runtime of a JVM process. While this error typically indicates an exhaustion of the java heap, a number of other potential causes exist which must not be overlooked when troubleshooting.

Note! Encountering an `OutOfMemoryError` is a critical event for a system, because it can break any function which encountered it, and as a result can cause the software to be in an unpredictable state after encountering the error. This error is one that can not be caught and handled gracefully.

Possible Causes and Resolutions

The following problems are known to cause an the OutOfMemoryError to be thrown.

Exhaustion of Old Generation in Java Heap

This is the most typical cause of an OutOfMemoryError. In this case, the java application requested more memory than was available in the java heap, and the amount required could not be made available even after garbage collection.

Some garbage collectors (eg. the throughput collector) will throw an OutOfMemoryError if too much time is being spent doing garbage collection. For example, if the JVM is spending more than 98% of the total time doing garbage collection and is recovering less than 2% of the heap, it will throw an out-of-memory exception.

To diagnose this problem, consider the following:

- Enable verbose GC logging (consult the documentation specific to your JVM vendor) Examine GC logs to see if garbage collections are unable to free up space. There are various GC tools available to analyze GC logs. Some options include:
 - HPjTune
Tool focused on the analysis and tuning of garbage collection behavior.
 - Diagnostic Tool for Java Garbage Collector (IBM JVM Only)
Tool from IBM focused on the analysis and tuning of garbage collection behavior.
 - jvmstat
Tool focused more on the analysis of *[garbage collection]* behavior. Lightweight enough to be run against a production environment.
 - GC Analyzer Graph
A utility to read the GC time and the number of GC cycles for Sun JVM generated gc logs..
- For the Integration Server analyze the stats.log file to see if available memory is shrinking consistently over time.

To resolve this problem, try increasing the java max heap using -Xmx.

Exhaustion of Permanent Generation

This generation has its own configured limit, and can be exhausted even when the overall Java Heap has plenty of space available. The JVM default size for this generation is very small for large server-side applications, and must typically be increased to avoid this error.

The error seen usually looks like

```
java.lang.OutOfMemoryError: PermGen space
```

The best way to test for exhaustion of the Permanent Generation is to obtain GC logging at a level that will show you the size of the permanent generation over time. If the permanent generation is at or near 100% utilization, then the environment is at risk for encountering an OutOfMemoryError due to this situation. The permanent generation should be increased to a sufficient level to accommodate all allocations that are made to it.

The following parameters can be used:

- "-XX:PermSize="
Initial Permanent Generation Size
- "-XX:MaxPermSize="
Max Permanent Generation Size

Exhaustion of OS System Memory

This situation might happen if your JVM goes through a heap expansion event, but the OS does not have sufficient memory resources to satisfy that request. A heap expansion event will only occur when the JVM MIN HEAP setting is configured to be less than the JVM MAX HEAP setting.

Some JVMs will exit or crash during this event.

Lack of Sufficient Native Memory

This problem is more likely to occur on 32-bit systems. If the JVM heap size is set too big, there may not be enough native memory remaining for your application to run.

When an application runs out of native memory, the JVM heap may still have plenty of space available. To address this problem, users can reduce their JVM Max heap size. By doing so, you will allow more space for the native memory.

Lack of Sufficient Thread Resources on JVM or on OS

In this case, the `java.lang.OutOfMemoryError` is caused by excessive threads being created. There are various reasons why the number of threads is excessive.

This error will typically reveal itself with the following message:

```
OutOfMemoryError: unable to create new native thread
```

There are a few things to do if you encounter this exception:

- Use the `lsof -p PID` command (Unix platforms) to see how many threads are active for this process.
- Determine if there is a maximum number of threads per process defined by the operating system. If the limit is too low for the application, try raising the per-process thread limit.
- Examine the application code to determine if there is code that is creating threads or connections (such as LDAP connections) and not destroying them. You could dump the Java threads to see if there are an excessive number has been created.

Sun JVM 1.5 and Windows family 32bit OSs

With this configuration the more memory you give to the JVM the more likely you are to get `java.lang.OutOfMemoryError: unable to create new native thread` exceptions when you have many threads.

Sun's JDK 1.4 allocates about 256K of address space per thread. With JDK 1.5, it is now allocating about 1M of address space per thread. Those numbers come from an assumption of a 2G address space per process under Windows, and the following statistics for the maximum number of threads you can create:

JDK 1.4:

- `-Xmx750` = 4580 threads.
- `-Xmx1000` = 3608 threads.
- `-Xmx1500M` = 1663 threads

JDK 1.5:

- `-Xmx750M` = 1129 threads
- `-Xmx1000M` = 880 threads
- `-Xmx1500M` = 384 threads

The following formula gives a decent estimate for the number of threads you can create:

$$(\text{MaxProcessMemory} - \text{JVMMemory} - \text{ReservedOsMemory}) / (\text{ThreadStackSize}) = \text{Number of threads}$$

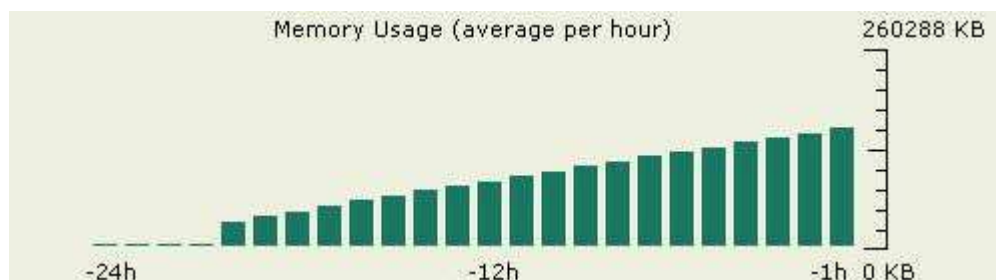
The possible workarounds for this issue are:

1. Reduce the thread stack size.
2. Lower the most as possible the max heap size.
3. Try the 3gb switch from Microsoft (<http://www.Microsoft.com/whdc/system/platform/server/PAE/PAEmem.msp>).
4. Change operating system (go for a 64bit Windows or a Unix/Linux system).

In this case, the `java.lang.OutOfMemoryError` is caused by excessive threads being created. There are various reasons why the number of threads is excessive. This may be caused by an exhaustion of JVM heap resources, or by encountering an OS-imposed restriction on the number of threads and/or processes which can be allocated.

FAQ

Memory usage on the Integration Server is (slowly) increasing and never released. Is this a memory leak? And is there anything we can do to address this?



This is most likely NOT a memory leak. It is most likely increasing because the JVM is not performing Full Garbage Collection - which generally doesn't happen until you actually need memory. So if your application is not using much memory, there is no need for the JVM to do a major GC. The JVM will only really clean up the memory at a later point in time, when it needs to.

If you let your example graph run for a while longer, you will eventually see that the JVM will do a "major GC event" and the memory on the IS stats page will go back down (and then begin the steady crawl back up). This is absolutely normal, and just has to do with how the JVM manages memory using periodic garbage collection.

The best way to determine if you are running out of memory is to trace the JVM's "Garbage Collection" behavior.

All JVMs have options to turn on tracing of this information, and then you can then load it into an analysis tool such as HPjTune or GCViewer to see the real memory usage graphs. It should only concern you if you see that the "Old Generation" is growing over time, and if "Full/Major GC events" are not cleaning up enough memory.

Platform-wide - Network

When troubleshooting network-related problems with the webMethods product suite, you can use the following techniques and utilities (often core OS-provided executables) to help diagnose the cause.

ipconfig / ifconfig

The ipconfig command (on Windows) or ifconfig command (on UNIX) provides the simplest means to list network interfaces, including IP addresses for a given host. For example, on Windows:

```
D:\>ipconfig

Windows IP Configuration

...
Ethernet adapter Interface 1:
Connection-specific DNS Suffix . : 
IP Address. . . . . : 192.168.1.102
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1

Ethernet adapter Interface 2:
Media State . . . . . : Media disconnected
```

Using ipconfig /all gives the full details of each interface (including DHCP settings and DNS server).

ping

The ping utility is the simplest means to test network connectivity to a given IP address (either local or remote). For example, on Windows:

```
D:\>ping empower.softwareag.com

Pinging empower.softwareag.com [193.26.193.147] with 32 bytes of data:

Reply from 193.26.193.147: bytes=32 time=19ms TTL=54
Reply from 193.26.193.147: bytes=32 time=39ms TTL=54
Reply from 193.26.193.147: bytes=32 time=27ms TTL=54
Reply from 193.26.193.147: bytes=32 time=28ms TTL=54

Ping statistics for 193.26.193.147:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 19ms, Maximum = 39ms, Average = 28ms
```

Occasionally, corporate networks will be configured to only allow certain types of traffic (for example, HTTP or FTP) and a ping request will fail. But typically, if a tcp-level ping works, then the foundation is there for other communications protocols to work.

netstat

The netstat command can serve two primary purposes:

- It lists the various port numbers in use on the current host (for example, 5555, 6849) and their current state (for example, LISTENING); and
- It lists the various connections to and from this host and its various ports

For example, the below output indicates the webMethods Integration Server (5555) and Broker Server (6849) are listening on this host, and also that several connections are currently established to the Broker Server from various remote hosts (10.128.48.*).

```
D:\>netstat -an
Active Connections
Proto Local Address Foreign Address State
...
TCP 0.0.0.0:5555 0.0.0.0:0 LISTENING
TCP 0.0.0.0:6849 0.0.0.0:0 LISTENING
...
TCP 10.128.48.229:1076 127.0.0.1:6849 ESTABLISHED
TCP 10.128.48.226:1076 127.0.0.1:6849 ESTABLISHED
TCP 10.128.48.232:1076 127.0.0.1:6849 ESTABLISHED
...
```

telnet

Like the ping command, the telnet command can be used to test network connectivity to a given IP address and port on that IP address (either local or remote).

For example, from a Windows host, the below telnet response indicates that the local telnet client could not find a listening port 1520 on host myOracleHost:

```
D:\>telnet myOracleHost 1520
Connecting To myOracleHost...Could not open connection to the host, on port 1520: Connect failed
```

But if that telnet command was pointed to a proper port number (for example, 1521), the console would clear (indicating the telnet session was established, and the client is awaiting the user to key in commands). In this sense, we are using telnet like a ping, but to a specific port number.

Ports Bound to Specific Interfaces

The netstat output shown above indicates ports listening on all interfaces for a given host (0.0.0.0). This means a request to 0.0.0.0:5555 or localhost:5555 or interface1:5555 would all reach this Integration Server. However, many of the webMethods applications (for example, Integration Server, My webMethods Server) have the ability to be bound to a single network interface (such as interface1). Keep this distinction in mind when configuring clients that connect to these server applications.

Source IP Address

Every tcp request will have a source IP address (that is, a "from" address) included on it when it reaches its destination. This address most likely will not be the IP address shown when issuing an ipconfig command on the originating host. Instead it will be an "outbound" address, set by the network's Default Gateway or outbound proxy server. This information may be useful when the server-side network or application (such as Integration Server) does IP-level filtering. There are public sites on the internet (for example, [<http://whatismyip.com>]) to which a browser can be pointed at, to provide this "outbound" IP address.

Intermittent Behavior

Anytime there is an intermittent behavior (for example, a connection cannot be established, an established connection times out, or a request cannot find a desired resource), there could be numerous factors involved. But some common causes to keep in mind or investigate are:

- Timeout settings throughout the connection: settings on the client application, on the sending application, on components in between (such as firewall, proxy servers, load balancers) the two, etc.
- Request data sizes: consider the possibility that larger requests are causing a timeout in one of the aforementioned components.
- Load-balancer presence: it could be that requests that are routed to one server node always work, but those routed to a second server node do not. To a client, this looks like an "intermittent failure".

Network/Packet Traces

A network/packet trace may be necessary in extreme cases. They should be collected by a network or systems administrator at a customer site (with proper filters applied), and analyzed by the customer and / or Software AG support.

One third-party tool that seems prevalent among customers is Wireshark (formerly Ethereal).

Platform-wide - SSL

Troubleshooting SSL typically involves dealing with problems that come up during the SSL handshake. Most problems are related to certificates, chain verification, or authentication. SSL configuration can be confusing and non-intuitive to those unfamiliar with it at a low level. As a result, most reported SSL problems are the result of misconfiguration on either the SSL client or SSL server.

General Troubleshooting

SSL Handshake Debugging

- Integration Server - the configuration property **watt.ssl.iaik.debug=true** can be set on an Integration Server to capture an SSL handshake when IS acts as either a client or server in the handshake.
 - Note that in older versions (6.5 and earlier) of IS, the property `watt.net.ssl.debug=true` was used for inbound tracing- this property is no longer valid (in 7.1.x and later).
 - Changing the value for `watt.ssl.iaik.debug` requires you to restart the Integration Server.
 - The debug output is written to STDOUT, and therefore can likely be found on the process's console or `nohup.out` file (on Unix).
 - Examples of the trace are below:
(outbound == Integration Server is https client)

```
ssl_debug(1): Starting handshake (iSaSiLk 3.03)...
ssl_debug(1): Sending v2 client_hello message, requesting version 3.1...
ssl_debug(1): Received v3 server_hello handshake message.
ssl_debug(1): Server selected SSL version 3.1.
ssl_debug(1): Server created new session C5:BE:45:57:CC:9D:54:0E...
ssl_debug(1): CipherSuite selected by server: SSL_RSA_WITH_RC4_128_MD5
ssl_debug(1): CompressionMethod selected by server: NULL
ssl_debug(1): Received certificate handshake message with server certificate.
ssl_debug(1): Server sent a 1024 bit RSA certificate, chain has 2 elements.
ssl_debug(1): Received server_hello_done handshake message.
ssl_debug(1): Sending client_key_exchange handshake message (1024 bit)...
ssl_debug(1): Sending change_cipher_spec message...
ssl_debug(1): Sending finished message...
ssl_debug(1): Received change_cipher_spec message.
ssl_debug(1): Received finished message.
ssl_debug(1): Session added to session cache.
ssl_debug(1): Handshake completed, statistics:
ssl_debug(1): Read 1547 bytes in 3 records, wrote 182 bytes in 3 records.
```

(inbound == Integration Server is https server)

```
ssl_debug(1): Starting handshake (iSaSiLk 3.03)...
ssl_debug(1): Received v2 client hello message.
ssl_debug(1): Client requested SSL version 3.1, selecting version 3.1.
ssl_debug(1): Creating new session C5:BE:45:57:CC:9D:54:0E...
ssl_debug(1): CipherSuites supported by the client:
ssl_debug(1): SSL_RSA_WITH_RC4_128_MD5
ssl_debug(1): SSL_RSA_WITH_RC4_128_SHA
ssl_debug(1): SSL_RSA_WITH_IDEA_CBC_SHA
ssl_debug(1): SSL_RSA_WITH_DES_CBC_SHA
ssl_debug(1): SSL_RSA_WITH_3DES_EDE_CBC_SHA
ssl_debug(1): SSL_RSA_EXPORT_WITH_RC4_40_MD5
ssl_debug(1): SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
ssl_debug(1): SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
ssl_debug(1): SSL_DH_DSS_EXPORT_WITH_DES40_CBC_SHA
ssl_debug(1): SSL_DH_DSS_WITH_DES_CBC_SHA
ssl_debug(1): SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA
ssl_debug(1): SSL_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
ssl_debug(1): SSL_DH_RSA_WITH_DES_CBC_SHA
```

```

ssl_debug(1): SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA
ssl_debug(1): SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
ssl_debug(1): SSL_DHE_DSS_WITH_DES_CBC_SHA
ssl_debug(1): SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
ssl_debug(1): SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
ssl_debug(1): SSL_DHE_RSA_WITH_DES_CBC_SHA
ssl_debug(1): SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
ssl_debug(1): SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
ssl_debug(1): SSL_DH_anon_WITH_RC4_128_MD5
ssl_debug(1): SSL_DH_anon_WITH_DES_CBC_SHA
ssl_debug(1): SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
ssl_debug(1): SSL_RSA_WITH_NULL_MD5
ssl_debug(1): SSL_RSA_WITH_NULL_SHA
ssl_debug(1): CompressionMethods supported by the client:
ssl_debug(1): NULL
ssl_debug(1): Sending server_hello handshake message.
ssl_debug(1): Selecting CipherSuite: SSL_RSA_WITH_RC4_128_MD5
ssl_debug(1): Selecting CompressionMethod: NULL
ssl_debug(1): Sending certificate handshake message with server certificate...
ssl_debug(1): Sending server_hello_done handshake message...
ssl_debug(1): Received client_key_exchange handshake message.
ssl_debug(1): Received change_cipher_spec message.
ssl_debug(1): Received finished message.
ssl_debug(1): Sending change_cipher_spec message...
ssl_debug(1): Sending finished message...
ssl_debug(1): Session added to session cache.
ssl_debug(1): Handshake completed, statistics:
ssl_debug(1): Read 182 bytes in 3 records, wrote 1547 bytes in 3 records.
ssl_debug(1): Shutting down SSL layer...
ssl_debug(1): Sending alert: Alert Warning: close notify
ssl_debug(1): Read 219 bytes in 1 records, 198 bytes net, 198 average.
ssl_debug(1): Wrote 3342 bytes in 1 records, 3321 bytes net, 3321 average.
ssl_debug(1): Closing transport...

```

- My webMethods Server - the jvm parameter **-Djavax.net.debug=ssl,handshake** can be added to the My webMethods Server startup command, by modifying `<webMethods_install_directory>/MWS/bin/mws.bat` (or `.sh` on Unix) as seen here:

```

set JAVA_OPTIONS=%JAVA_OPTIONS% -Dserver.name=%SERVER_NAME% -Djava.awt.headless=true -
Djavax.net.debug=ssl,handshake

```

...which will produce debug information like below (listing trusted Certificate Authorities):

```

...
adding as trusted cert:
Subject: CN=DigiCert High Assurance EV Root CA, OU=www.digicert.com, O=DigiCert Inc, C=US
Issuer: CN=DigiCert High Assurance EV Root CA, OU=www.digicert.com, O=DigiCert Inc, C=US
Algorithm: RSA; Serial number: 0x2ac5c266a0b409b8f0b79f2ae462577
Valid from Thu Nov 09 19:00:00 EST 2006 until Sun Nov 09 19:00:00 EST 2031
adding as trusted cert:
Subject: CN=Baltimore CyberTrust Code Signing Root, OU=CyberTrust, O=Baltimore, C=IE
Issuer: CN=Baltimore CyberTrust Code Signing Root, OU=CyberTrust, O=Baltimore, C=IE
Algorithm: RSA; Serial number: 0x20000bf
Valid from Wed May 17 10:01:00 EDT 2000 until Sat May 17 19:59:00 EDT 2025
...

```

Certificate Formats

The supported certificate formats for the core webMethods products are:

- Integration Server:
 - 7.x - DER / PEM for all server and partner certificates
 - 8.x - various formats (such as JKS and PKCS12) for server keystores and truststores, and DER / PEM for mapped partner certificates
- Broker Server and clients:
 - 7.x and 8.x - various formats (such as JKS and PKCS12) for all keystores and truststores
- MWS
 - 7.x and 8.x - various formats (such as JKS and PKCS12) for all keystores and truststores

Viewing Certificate Details

Depending on the format, here are some of the tools Software AG engineers will typically use to help solve certificate issues:

- DER - OpenSSL on any platform or Internet Explorer on Windows
- PEM - OpenSSL on any platform or Internet Explorer on Windows (assuming you rename to .DER, though it is still technically PEM format)
- JKS - java keytool utility
- PKCS12 - OpenSSL or a browser (assuming you import it into the browser first)

The information that is typically useful when inspecting a certificate is the subject, issuer, and expiration date.

Enabling SSL / HTTPS on My webMethods Server

The general process is:

1. In the My webMethods Server, navigate to Administration > System Settings > Cluster Settings > Advanced or Cluster Configuration > specify an HTTPS port #, then shutdown the My webMethods Server.
2. The file <webMethods_install_directory>\MWS\server\default\config\jetty.xml is already configured to run with the sample keystore "demo.keystore" provided by Software AG:

```
...
<Set name="Keystore"><SystemProperty name="server.home"
default="." />/config/<SystemProperty name="keystore.file" default="demo.keystore" /></Set>
<Set name="Password">password</Set>
<Set name="KeyPassword">password</Set>
...
```

3. Restart the My webMethods Server. The demo.keystore (JKS) referenced above will be used to hand out the server certificate chain.
To determine which truststore (JKS) is used:
 - if the Central Configuration UIs are not installed on the My webMethods Server, the HTTPS port will use a trust list based on:

- the underlying JVM's cacerts file (for example, <SoftwareAG>\jvm\win150\jre\lib\security\cacerts); and
 - the subject and issuer from the demo.keystore;
 - if the Central Configuration UIs are installed on the My webMethods Server, the HTTPS port will use a trust list based on:
 - the Glue truststore file (by default <SoftwareAG>\MWS\server\default\config\glue\glueTrustStore.jks – non-default can be specified in <SoftwareAG>\MWS\server\default\config\glue\glueSSLConfiguration.xml); and
 - the subject and issuer from the demo.keystore;
4. If you wish to use a different keystore (other than the demo.keystore), you can modify the above 3 lines in the jetty.xml. The xml syntax for the first line changes to:

```
...
<Set name="Keystore"><SystemProperty name="server.home"
default="." />/config/user.keystore</Set>
<Set name="Password">password1</Set>
<Set name="KeyPassword">password2</Set>
...
```

Note: it is important to have the correct storePass and keyPass specified. If the storePass is wrong, you will receive the following error during startup of the My webMethods Server:

```
"java.io.IOException: Keystore was tampered with, or password was incorrect"
```

...and if the keyPass is wrong, you will receive the following error during startup of the My webMethods Server:

```
"java.security.UnrecoverableKeyException: Cannot recover key"
```

You can verify the storePass and keyPass (for a given key pair in a keystore) using the following keytool command. If either is wrong, you will see the same two errors as above.

```
keytool -keypasswd -keystore user.keystore -storepass password1 -alias myAlias -keypass
password2 -new password2
```

OpenSSL

OpenSSL is a powerful, free, and widely used tool that is useful for troubleshooting SSL problems.

This section is a quick reference of the most useful OpenSSL commands.

Certificates and Private Keys

This section lists useful operations that can be performed on Certificates and Private Keys.

View Certificate Details

To view a PEM-format certificate:

```
openssl x509 -in certificate.pem -inform PEM -text
```

To view a DER-format certificate:

```
openssl x509 -in certificate.der -inform DER -text
```

View the purpose fields of a certificate.

To view the purpose fields of a given PEM-formatted certificate:

```
openssl x509 -in mycert.pem -purpose
```

View Raw dump of a DER/PEM format file

This can be used to help figure out what kind of file you are looking at:

```
openssl asn1parse -in priv9.der -inform DER
```

Create a DER-format Private Key and Certificate

Follow these steps, in order:

- Generate 1024-bit Private Key:

```
openssl genrsa -out private_key.pem 1024
```

Additional options are to use `-des3` option to password protect the key, and and to use `-rand some_file` option to introduce additional sources of entropy to the pseudorandom number generator.

- Use that Private Key to Generate a Certificate Signing Request (CSR):

```
openssl req -new -key private_key.pem -out certificate_request.csr
```

- Use the CSR to Create and Sign a Certificate

```
openssl x509 -req -days 360 -in certificate_request.cs -signkey private_key.pem -out certificate.cert
```

- There is an easy way to handle certificate related actions. The OpenSSL team provides a nice PERL script located under `<OpenSSL>/bin/CA.pl`. This scripts encapsulates calls to OpenSSL commands. To use this script you'll need to install ActivePerl. This script can be modified to suit your needs. Below are some of the possible commands:
 - `CA.pl -newca ...` will setup the right stuff
 - `CA.pl -newreq ...` will generate a certificate request
 - `CA.pl -sign ...` will sign the generated request and output

Create a PKCS#12 file

To create a PKCS#12 file:

- Create (or locate an existing) DER-format private key and certificate (instructions above).
- Follow the instructions below to convert the DER-format certificate and private key to PKCS#12 format.

Test if Certificate and Private Key Actually Match

To test whether a certificate and a private key actually match, you can compare their "modulus". The modulus must be the same for a matching certificate and private key pair.

1. **Get the modulus for the certificate:**

To view the modulus for a certificate, run the following command:

```
openssl x509 -in certificate_filename.der -inform D -modulus -noout
Modulus=A5F547BE7C53BE4E3FA272757585672A0C0DA6CF6A90925DC9A9792FF660B509A26321CA
64C8D12428127C93F26514E05F91F6A5CABCD5AF0C49F5659CD220C5DC68D226FF5F5BBFD3C3C86
9E495B25A6A1E13BDDFF24BDBC00641E52514F7F061D02BCB71A7A13936C5405C0D06D52D069A8225
161F39C5DAF538271327B9B7
```

2. Get the modulus for the private key:

To view the modulus for a private key, run the following command:

```
openssl rsa -in private_key_filename -inform DER -modulus -noout

Modulus=A5F547BE7C53BE4E3FA272757585672A0C0DA6CF6A90925DC9A9792FF660B509A26321CA
64C8D12428127C93F26514E05F91F6A5CABCD5AF0C49F5659CD220C5DC68D226FF5F5BBFD3C3C86
9E495B25A6A1E13BDDF24BDBC00641E52514F7F061D02BCB71A7A13936C5405C0D06D52D069A8225
161F39C5DAF538271327B9B7
```

If you get an error such as ...

```
5496:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wrong
tag:.\crypto\asn1\tasn_dec.c:1294:
5496:error:0D06C03A:asn1 encoding routines:ASN1_D2I_EX_PRIMITIVE:nested asn1
error:.\crypto\asn1\tasn_dec.c:830:
5496:error:0D08303A:asn1 encoding routines:ASN1_TEMPLATE_NOEXP_D2I:nested asn1
error:.\crypto\asn1\tasn_dec.c:749:Field=n, Type=RSA
5496:error:0D09A00D:asn1 encoding routines:d2i_PrivateKey:ASN1
lib:.\crypto\asn1\d2i_pr.c:99:
```

... then you may have to convert the private key using one of the pkcs commands such as pkcs8.

For example:

```
openssl pkcs8 -in private_key.der -nocrypt -inform DER | openssl rsa -modulus -noout

Modulus=A32A40135AD04E1659F5E3E88A262207DD5A18637EBC28B1D129D154FBF0A942E06652C6CD6929D8DCC
1108EE874EC6999A8767E39B0796E
08F7BA3CE7BF7CC05B0E530D830F781D279AD691DF70A043C3E1B4FC060C97431D0C899EAB369DAF2FB2F47923B
076F9592CD74FF6D7E879BB1912CC
8E9987BC3E8EE8A8C7121685197C543CE6E0DD96CA471A4E25FDE8C21FD57674D2CF0C8825869F26BD3CF25B8A3
6CE1D11B9123D9DBF5925138A130D
C13ED87B6DF01AF8ABFCF9E5E31967C9
```

3. Compare the hex values for each modulus:

The hex numbers should match. If they do not match, then the private key does not match the certificate, and thus RSA operations performed using this pair would expectedly fail.

Convert format of Certificates

From PEM-format to DER-format:

```
openssl x509 -in input_cert.pem -inform PEM -out output_cert.der -outform DER
```

From DER-format to PEM-format:

```
openssl x509 -in input_cert.der -inform DER -out output_cert.pem -outform PEM
```

Convert format of Private Keys

From PEM-format to DER-format:

```
openssl rsa -in input_key.pem -inform PEM -out output_key.der -outform DER
```

From DER-format to PEM-format:

```
openssl rsa -in input_key.der -inform DER -out output_key.pem -outform PEM
```


If you get an error such as this during DER -> PEM conversion...

```
5496:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wrong
tag:.\crypto\asn1\tasn_dec.c:1294:
5496:error:0D06C03A:asn1 encoding routines:ASN1_D2I_EX_PRIMITIVE:nested asn1
error:.\crypto\asn1\tasn_dec.c:830:
5496:error:0D08303A:asn1 encoding routines:ASN1_TEMPLATE_NOEXP_D2I:nested asn1
error:.\crypto\asn1\tasn_dec.c:749:Field=n, Type=RSA
5496:error:0D09A00D:asn1 encoding routines:d2i_PrivateKey:ASN1 lib:.\crypto\asn1\d2i_pr.c:99:
```

... then you may have to convert the private key using one of the pkcs commands such as pkcs8. For example:

```
openssl pkcs8 -in input_key.der -inform DER -out output_key.pem -outform PEM
openssl pkcs8 -nocrypt -in input_key.der -inform DER -out output_key.pem -outform PEM
openssl asn1parse -in input_key.der -inform DER
```

Convert DER-format certificate and private key to PKCS#12

To convert a DER-format certificate and private key into a PKCS#12 file:

```
openssl pkcs12 -export -in servercert.pem -inkey serverpk.pem -CAfile rootca.pem -name "server" -
out server.p12
Enter Export Password:
Verifying - Enter Export Password:
```

Export certificates and keys from a PKCS#12 file

Integration Server can not import PKCS#12 directly. The following steps can be used to export the certificates and keys in a format Integration Server can use. Note that this procedure only covers PKCS#12 files with a single keypair. More complex PKCS#12 files require additional intermediate steps.

Export the client certificate from a PKCS#12 certificate

```
openssl pkcs12 -in alice.p12 -clcerts -nokeys > alicecert.pem
```

Export the CA certificates from a PKCS#12 file

```
openssl pkcs12 -in alice.p12 -cacerts -nokeys > rootca.pem
```

Export the private key from a PKCS#12 file

```
openssl pkcs12 -in alice.p12 -nocerts -nodes > alicepk.pem
```

SSL

Test an SSL Handshake using OpenSSL as the client

To test an SSL connection to server and to display the server's certificate chain:

```
openssl s_client -connect hostname:port -verify 6 -showcerts -state -msg
```

Where hostname:port is the hostname and port of the server you are testing.

The "-state" option will show each change in SSL handshake state, and thus will show you at which point in the handshake it is failing.

You can compare the output to a known-to-be-working server which presents a correct chain, and where the SSL handshake is successful (For example, empower.softwareag.com:443).

Test an SSL Handshake using OpenSSL as the server

OpenSSL can be used to emulate an SSL server, simulating an external partner SSL server.

Note: On some versions of OpenSSL, the certificate and key must be in PEM format - use the DER -> PEM conversion instructions listed in this article if needed.

Create an SSL server which will request Client Authentication:

```
openssl s_server -accept <port> -cert cert.pem -key key.pem -verify <arg>
```

Create an SSL server which will require Client Authentication:

```
openssl s_server -accept <port> -cert cert.pem -key key.pem -Verify 3
```

Verify Certificate Chain (Local)

To verify a local certificate chain ("local" means you have all the certificate and CA files on your local filesystem):

```
openssl verify -verbose -CAfile ca.pem cert.pem
```

where ca.pem is a PEM-encoded CA certificate, and cert.pem is a PEM-encoded certificate. This test will verify if cert.pem was signed by ca.pem.

S/MIME

The smime command handles S/MIME mail. It can encrypt, decrypt, sign and verify S/MIME messages.

Viewing the Signature in an S/MIME document

To view the S/MIME signature:

(smime.txt is a saved S/MIME email message, and the command output follows)

```
openssl smime -pk7out -in smime.txt | openssl pkcs7 -text -noout -print_certs
Certificate:
Data:
Version: 1 (0x0)
Serial Number: 1115360292 (0x427b0c24)
Signature Algorithm: sha1WithRSAEncryption
Issuer: CN=Meica
Validity
Not Before: May 6 06:18:12 2005 GMT
Not After : May 6 06:18:12 2006 GMT
Subject: CN=Meica
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:d9:99:12:5c:fa:6f:ec:ec:64:b6:99:75:43:19:
c6:a3:5c:8b:04:62:3c:67:40:db:ba:35:3e:25:31:
5a:b0:4b:0e:e4:7c:bd:ba:b7:83:30:f6:83:b0:1f:
59:47:9e:97:1e:c5:99:8c:3e:4c:11:8d:86:2e:1b:
30:0a:ab:9f:a5:49:a6:8e:02:af:cf:85:52:55:3c:
92:f1:a9:ab:68:bc:63:26:9f:82:e6:1d:21:70:ba:
3b:72:51:4d:90:24:51:5e:0c:5e:66:0e:c0:65:9d:
43:a1:49:66:33:a8:a9:7f:b6:85:d5:76:7d:52:78:
aa:93:dd:56:54:5c:9f:e5:37
Exponent: 65537 (0x10001)
Signature Algorithm: sha1WithRSAEncryption
89:45:13:98:c1:98:3c:40:de:06:f9:94:c2:3d:d4:aa:68:ce:
14:e8:d1:f2:07:6d:10:9d:b8:7d:6f:8c:d3:dc:76:12:7a:7b:
ae:57:6b:06:47:6f:c9:be:2d:74:26:7c:dd:09:b5:42:bd:24:
f6:7a:25:0d:30:99:1b:58:0f:69:53:d1:e3:c0:a8:39:ed:3b:
54:e6:3f:2b:d6:f6:99:4f:f1:1c:21:69:6e:33:de:32:fb:ab:
fe:a6:af:4a:ff:fb:57:04:bb:de:b9:e5:2e:90:3d:23:8c:31:
17:32:c6:95:50:3f:59:1c:7b:86:d8:d1:f0:26:cd:f4:e0:9f:
d6:15
```

Verify the Signature in an S/MIME document

A signed MDN is used as example S/MIME document here.

Let's assume the MDN is signed by the demo sender certificates provided with the webMethods EDIINT Module.

The sender certificates can be found under \IntegrationServer\packages\WmEDIINT\pub\demo.

Since the MDN doesn't contain the certificate information, the certificates have to be loaded into the openssl trusted directory.

These are the steps:

- Go to TN console to copy the signed MDN and save it as file in a directory. In this example, we call it mymdn.
- Extract the first certificate from senderCert.der and save it as a different der file which it is called senderRoot.der in this example.
- Then use the openssl command to convert the sender certificates and public key files from der to pem files. For example, senderRoot.der, senderCACert.der and senderCert.der.

```
openssl x509 -in senderRoot.der -inform DER -out senderRoot.pem -outform PEM
```

- Then make the sender certificates and public key trusted by openssl using the command

```
openssl x509 -in senderRoot.pem -addtrust serverAuth
```

senderRoot.pem is the certificate you want to put in the trust directory.

- Put all sender pem files in the openssl trust directory. Please see below for more information about the openssl trusted directory.
- Then run the following command to verify the signature:
- openssl smime -verify -CApath C:\OpenSSL\bin\PEM\ -CAfile C:\OpenSSL\bin\PEM\senderFirstPath.pem -certfile C:\OpenSSL\bin\PEM\senderCert.pem -signer C:\OpenSSL\bin\PEM\senderCACert.pem -in c:\mymdn

Configuration

OpenSSL Trusted Directory

When OpenSSL was built for your system, it was configured with a "Directory for OpenSSL files". (That's the --openssldir option passed to the configure script.) This is the directory that typically holds information about certificate authorities your system trusts.

The trick to find the trusted directory is to find the cert.pem file.

Within that directory and a subdirectory called certs, you're likely to find one or more of three different kinds of files.

- A large file called cert.pem, an omnibus collection of many certificates from recognized certificate authorities like VeriSign and Thawte.
- Some small files in the certs subdirectory named with a .pem file extension, each of which contains a certificate from a single CA.

Some symlinks in the certs subdirectory with obscure filenames like 052eae11.0. There is typically one of these links for each .pem file.

Common Errors

Below is a list of errors that are commonly seen in Integration Server during SSL communication, with some details and potential resolutions for each. Keep in mind there may be multiple causes (and resolutions) for these error messages, and the exact text of the messages themselves may vary slightly depending on a number of factors (such as JVM version, Security provider libraries in use, or version of those security libraries).

Server Certificate Rejected By ChainVerifier

The Integration Server (acting as a client) was presented a certificate chain that it does not trust. Here are potential causes and corrections:

- The remote server's chain is not trusted
 - Add the root CA certificate (or at least some intermediate CA certificate from that chain) to the Integration Server's truststore in 8x (or Trusted CAs Directory in 7x)
- The remote server is presenting a chain with an expired CA certificate
 - The Integration Server can ignore this expired CA certificate (`watt.security.ssl.client.ignoreEmptyAuthoritiesList=true`), but can not ignore an expired subject certificate. Still, the best solution is for the remote server administrator to be notified and that server to be re-configured to not present any expired certificates.
- The remote server is presenting a chain that is out of order - per RFC 2246, a certificate chain that looks like this:

`subjectCert > intermediateCAcert > rootCAcert`

...can be handed out as:

`subjectCert > intermediateCAcert > rootCAcert` or
`subjectCert > intermediateCAcert`

...but not:

`subjectCert > rootCAcert`

- To use OpenSSL to test if a remote server's chain is out of order, use the following command:

```
openssl s_client -connect hostname:port -verify 6 -showcerts
```

This command will print each certificate in the order it was presented. Upon examination, the certificate's issuer should match the next certificate's subject, until the root is reached. The root CA certificate's issuer will equal its own subject.

- To resolve, the remote server administrator should re-configure the chain being handed out.

Record Version Mismatch: 02

This error could indicate that the client and server do not agree on the level of SSL to use (SSLv2, SSLv3, TLS). Integration Server can restrict what versions it uses (`watt.net.ssl.client.handshake.minVersion` and `watt.net.ssl.client.handshake.maxVersion`). To check if a remote server supports a particular version, use OpenSSL to run the following command (which actually checks if SSLv2 is supported):

```
> openssl s_client -connect hostname:443 -ssl2
```

If you get the below response, SSLv2 is disabled:

```
419:error:1407F0E5:SSL routines:SSL2_WRITE:ssl handshake failure:s2_pkt.c:428:
420:error:1406D0B8:SSL routines:GET_SERVER_HELLO:no cipher list:s2_clnt.c:450:
```

...otherwise, SSLv2 is enabled.

Peer sent alert: Alert Fatal: bad certificate

- This error can occur when the SSL server is presenting an empty CA list, which causes the client to not present its certificate at all. With `watt.ssl.iaik.debug=true` set, the client-side handshake output would tell us this about the remote server during the handshake:

```
ssl_debug(1): Starting handshake (iSaSiLk 3.03)...
ssl_debug(1): Sending v2 client_hello message, requesting version 3.1...
ssl_debug(1): Received v3 server_hello handshake message.
ssl_debug(1): Server selected SSL version 3.1.
ssl_debug(1): Server created new session 1A:7A:BD:14:83:CA:85:50...
ssl_debug(1): CipherSuite selected by server: SSL_RSA_WITH_RC4_128_MD5
ssl_debug(1): CompressionMethod selected by server: NULL
ssl_debug(1): Received certificate handshake message with server certificate.
ssl_debug(1): Server sent a 1024 bit RSA certificate, chain has 2 elements.
ssl_debug(1): Received certificate_request handshake message.
ssl_debug(1): Accepted certificate types: RSA, DSS
ssl_debug(1): Accepted certificate authorities:
ssl_debug(1): (empty list) // SSL Server is presenting an empty CA list
ssl_debug(1): Received server_hello_done handshake message.
ssl_debug(1): No client certificate available, sending empty certificate message... //
Thus, SSL Client responds with an empty certificate
ssl_debug(1): Sending client_key_exchange handshake message (1024 bit)...
ssl_debug(1): Sending change_cipher_spec message...
ssl_debug(1): Sending finished message...
ssl_debug(1): Received alert message: Alert Fatal: bad certificate
ssl_debug(1): SSLException while handshaking: Peer sent alert: Alert Fatal: bad certificate
// SSL Server rejects the empty certificate
ssl_debug(1): Shutting down SSL layer?
```

Integration Server can optionally be configured to ignore such a scenario and still present its own chain (`watt.security.ssl.client.ignoreEmptyAuthoritiesList=true`).

iaik.pkcs.PKCSException javax.crypto.BadPaddingException: Unknown blocktype

This error might be caused by a configuration where the private key and certificate do not match (that is, they are not a pair).

Certificate Authority (CA) is untrusted

For successful digital signature verification, for example as part of RosettaNet processing, ensure that the CA certificate of the certificate data on the message is in the Integration Server's truststore in 8.x, or in the Trusted CAs Directory in 7.1.x.

Information to Gather When Opening a Support Incident

To troubleshoot an SSL related issue, Software AG Global Support will need you to obtain the following information:

- Provide the information listed on the [Integration Server General Information](#) page.
- Inform if the Integration Server is acting like an SSL client or as an SSL server
- Output of OpenSSL command 'showcerts' (let this be done on the SSL client machine)
- Diagnostic_data.zip
- Screenshots of your Integration Server Administrator certificate settings

Platform-wide - Database

Database Scripts

For user convenience, the webMethods product suite offers a database creation/drop tool named the Database Component Configurator. The raw scripts that are run by the Configurator are stored in the `<SoftwareAG_install_directory>\common\db\scripts\<db_vendor>\<wM_component>\<script_version>\<operation>` directory.

For example, when you create the Archive database component on Oracle, the configurator runs the scripts and procedures in the `<SoftwareAG_install_directory>\common\db\scripts\oracle\archive\22\create` directory. (Note: Version 22 is just an example. The current version number varies across database components and releases.)

The configurator creates a log in the `<SoftwareAG_install_directory>\common\db\logs` directory for each session.

ConnectJDBC Driver (general)

The DataDirect Connect JDBC (CJDBC) drivers are used internally by various webMethods components such as Integration Server and My webMethods Server. These drivers are packaged so that they can only be used by the webMethods products. They can not be extracted and used in any other way.

To determine the CJDBC driver version and build being used by a given installation, open the .jar file and find the similarly named text readme file, or use the command below. (Note: The command below indicates CJDBC driver 3.6 build 0039.)

```
<SoftwareAG_install_directory>\common\lib\ext>jar tvf sl53_cj36.jar | grep fixes
34402 Mon Sep 10 17:06:14 EDT 2007 fixes_36_0039.txt
```

Provide this information to Software AG Global Support when reporting a database-related issue where any driver misbehavior is suspected.

ConnectJDBC Driver (trace files)

What you can do is to use the "spy driver" instead of the "normal DataDirect driver", in order to get more connection information. This spy driver is installed by default in the webmethods directory, inside the standard DataDirect driver jar file. The class is to be used only for troubleshooting purposes.

If you want to use it, please follow these steps:

1. On the JDBC Pools page, create a new Driver Alias Definition with the class name set to that of the DataDirect SpyDriver: `com.wm.dd.jdbcspy.SpyDriver`
2. Add a new Pool Alias Definition and choose the Driver Alias Definition you just created. Enter the database URL according to the following syntax:

```
jdbc:spy:{jdbc:wm:oracle://localhost:1521;SID=oracle};load=com.wm.dd.jdbc.oracle.OracleDriver;log=(file)C:\temp\spy.log
```

Between { } you need to place the original database URL. The log parameter points to the location where the trace file will be stored.

3. Change the relevant Functional Aliases to use this JDBC Pool Alias.
4. Restart the IS.

With this driver, it's possible to log all the database interaction between IS and database, but you need to be careful to use it in production.

Verbose Logging at Application Level

Some webMethods products have logging configurations that can be enabled at the application level to log verbose database activity. For example:

Monitor - all services that interface with the database from which Monitor reads logging data reside on Integration Server in the WmMonitor package. You can set the IS Admin > Settings > Logging > Default > Integration Server > 0120 Monitor Database = TRACE to log the raw SQL that is run by such services to the IS server.log.

For example:

```
...
2009-05-17 00:57:14 EDT [MOD.0120.0001T] in method SingleAccess.getList()
2009-05-17 00:57:14 EDT [MOD.0120.0020D] SQL statement: SELECT DISTINCT t9.ROOTCONTEXTID <...>
FROM WMSERVICE t9 <.....> ORDER BY t9.AUDITTIMESTAMP DESC
2009-05-17 00:57:14 EDT [MOD.0120.0021T] max rows for query: 0
```

My webMethods Server - you can send all SQL that MWS sends directly to its database by modifying the `sqlDebug log4j` category. Do the following:

- Log into MWS as sysadmin (default password is "manage").
- Go to Folders > Administrative Folders > Administration Dashboard > Analysis > Logging Configuration.

- Set the sqlDebug category to INFO (at time of this publication, the more verbose levels DEBUG/TRACE were not exposed via the user interface; you will need to modify the configuration file to do this).
- Stop the MWS server
- As of MWS 7.1.2, all configuration files are stored in the MWS_config table. To modify a config file, do the following:
 - Extract the config file from the database
(`<SoftwareAG_install_directory>/MWS/bin/mws getconfig log4j.override.properties`).
 - Modify the extracted file
(`<SoftwareAG_install_directory>/MWS/server/<instance>/config/log4j.override.properties`), setting the following:


```
...
appender.rootFile.threshold=TRACE
appender.rootOut.threshold=TRACE
...
category.sqlDebug.priority=TRACE
...
```
 - Optionally insert the config file back into the database
(`<SoftwareAG_install_directory>/MWS/bin/mws putconfig log4j.override.properties`).
- Restart the MWS server. (Note: This will generate very verbose output with raw SQL logged both to console and log file.)

Obtaining Database Diagnostics

It will occasionally be necessary to have a Database Administrator (DBA) analyze activity on the database server. This analysis might include, but not limited to, the following items:

- Sessions from known webMethods users
- Viewing locks
- Deadlock conditions
- Process-intensive queries.
- etc.

Platform-wide - Dr. Watson Logs

Collecting user-mode dumps with Windows 7, Windows Vista and Windows Server 2008

Starting with Windows Server 2008, Windows Vista with Service Pack 1 (SP1) and Windows 7, user-mode dumps can be collected and stored when an application crashed via the Windows Error Reporting (WER) feature.

This feature is not enabled by default, enabling this feature requires administrator privileges. To save these user mode memory dumps locally using Windows Error Reporting, create the following Registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps

Value Name = DumpType

Data type: REG_DWORD

Value Data = 1

Data Values Descriptions:

0 = Create a custom dump

1 = Mini dump

2 = Full dump

For more information about collecting user-mode dumps, visit the following Microsoft Web site:

[http://msdn.microsoft.com/en-us/library/bb787181\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb787181(VS.85).aspx)

About Dr. Watson

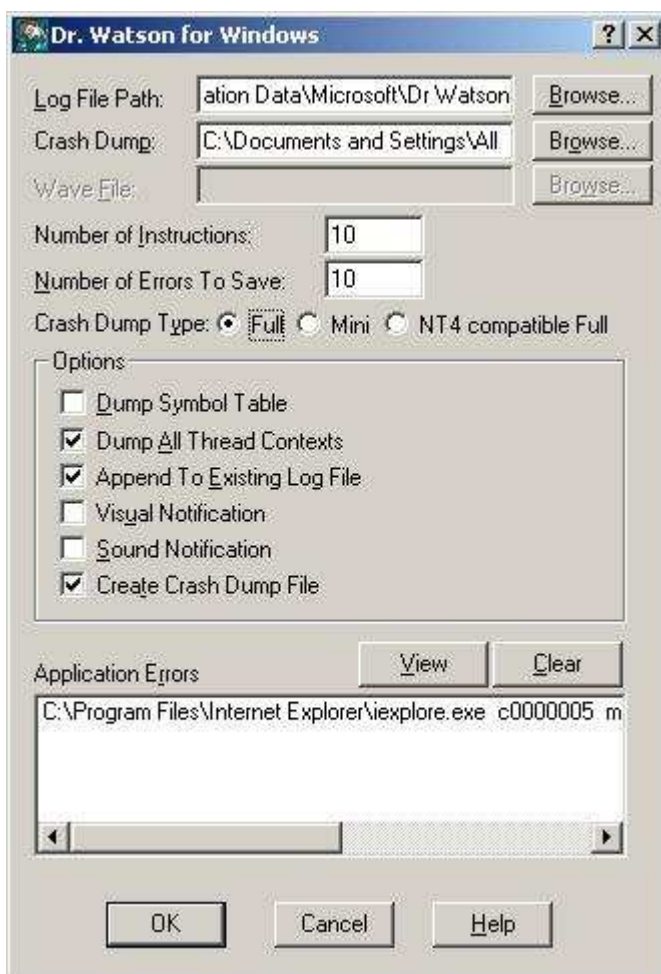
Windows releases prior to Windows 7, Windows Vista and Windows Server 2008 use Dr. Watson to create user-dumps. **Dr. Watson logs** are files created by Microsoft Windows whenever a significant process error is detected. Dr. Watson logs (typically named Drwtsn32.log) are written either as text or binary files depending upon your preferences. By default, Dr. Watson creates a crash dump file, which can be useful in debugging crashes of the webMethods Broker or the JVM that runs a Java-based webMethods product, like Integration Server or My webMethods Server.

Starting Dr. Watson

If a program error occurs, Dr. Watson may start automatically. You can also manually start Dr. Watson in either of the following ways:

- From Windows, click **Start > Run**, and then enter **drwtsn32**.
- From a command prompt, change to the root directory, and then enter **drwtsn32**. A dialog box will appear where you can choose the path where the log file and the core dump files will be saved. You can also choose different options to log.

If you have been using a program other than Dr. Watson as your default debugger, and you want to use Dr. Watson instead, go to the command prompt and enter the command **drwtsn32 -i** to start Dr. Watson. The -i switch causes the necessary changes to be made to the registry.



Index

acknowledge, 63, 74, 198
 ACL, 117
 Actional, 192, 195
 ActivePerl, 239
 ActiveSOA, 180, 181
 AdapterConnection, 89
 AdapterConnectionException, 88, 89
 AdapterException, 88, 89
 AdapterService, 95
 AdapterServiceException, 88, 111
 Administrator, 6, 9, 16, 17, 20, 22, 34, 38,
 41, 43, 46, 48, 49, 52, 53, 54, 55, 64, 65, 69,
 70, 73, 75, 77, 102, 109, 113, 114, 116, 117,
 118, 120, 155, 161, 169, 170, 172, 174, 175,
 177, 247, 250
 ADSI, 59
 agent, 190, 191, 192, 193, 194, 195, 196, 197,
 198
 AIX, 58, 81, 84, 221, 222, 223
 analysis, 201, 204, 226, 227, 230, 250
 Analytic Engine, 138, 139, 140, 141, 142
 AOP, 190, 197
 Apache, 181, 184, 189
 API, 46, 56, 57, 79, 103, 104, 106, 108, 186
 API4J, 186
 ART, 37, 44, 90, 110, 111
 AS/2, 125, 126
 AS400, 93
 audit, 13, 16, 21, 22, 32, 33, 69, 70, 152, 155,
 168, 191, 192, 198
 AuditLogManager, 33
 Authority, 247
 awbroker.cfg, 59, 82
 awbroker.pid, 61
 backend, 22, 65, 102, 143
 backup, 7, 159, 180
 basicauth.cfg, 59

batch, 61, 62, 97, 98
 blocked, 23, 66, 135
 bottleneck, 15, 16
 BPML, 173, 174
 Broker, 6, 36, 42, 43, 44, 45, 46, 47, 56, 57,
 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70,
 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82,
 83, 85, 86, 87, 138, 155, 163, 164, 165, 167,
 175, 201, 206, 207, 232, 236, 251
 broker.alert, 58
 broker.info, 58
 broker_admin.cnf, 14
 BrokerConfig.qs, 59, 60, 62
 BrokerConfig.qs.log, 60, 62
 BrokerConfig.qs.stor, 60, 62
 BrokerData.qs, 59, 60, 61, 62, 64, 65, 66, 67,
 68, 72
 BrokerData.qs.log, 60, 61, 62, 72
 BrokerData.qs.stor, 60, 61, 62, 64, 65, 66,
 67, 68, 72
 browser, 9, 20, 25, 55, 122, 140, 143, 144,
 145, 161, 173, 189, 192, 205, 207, 233, 236
 buffer, 92, 219
 build, 11, 26, 56, 112, 114, 122, 143, 146,
 182, 200, 201, 248
 C/C++, 31
 CA, 235, 239, 241, 242, 244, 245, 246, 247
 cache, 39, 41, 54, 116, 158, 199, 211, 216,
 234, 235
 CAF, 143, 144, 145, 146
 catalog, 139
 category, 249, 250
 CCSID, 111
 CentraSite, 180, 181, 182
 certificate, 128, 234, 235, 236, 238, 239,
 240, 241, 242, 243, 244, 245, 246, 247
 chain, 128, 130, 234, 236, 242, 245, 246
 ChainVerifier, 245
 channel, 94
 character, 79
 chmod, 182

chown, 182
 CJDDBC, 114, 248
 ClassCastException, 154
 classpath, 40, 41, 73, 74, 75, 86, 87, 107, 115, 151, 186
 ClearQueue.java, 75
 client.policy, 211
 ClientGroup, 73, 74, 75
 cluster, 7, 12, 39, 40, 41, 51, 52, 53, 54, 65, 66, 67, 92, 112, 115, 157, 169, 170, 199, 200, 217
 codegen.cnf, 14
 Coherence, 39, 41, 54
 collector, 227
 column, 56, 65, 71, 95, 118, 141, 155, 212, 213, 222, 223, 224
 common, 10, 18, 22, 32, 40, 41, 42, 44, 55, 57, 68, 73, 74, 75, 86, 87, 114, 121, 131, 132, 134, 143, 147, 149, 176, 181, 190, 202, 209, 221, 233, 248
 communities, 7, 133
 compat-libstdc++, 207
 compile, 10, 73, 74, 75, 86
 compiler, 10, 26
 compliant, 44, 132
 concurrent, 55, 68, 77, 159
 ConnectionManager, 34
 ConnectJDBC, 248, 249
 console, 34, 41, 120, 147, 148, 161, 190, 198, 204, 208, 219, 220, 232, 234, 244, 250
 Content-Type, 129, 130
 contiguous, 72
 conversation, 39, 166
 ConversationID, 167
 conversion, 79, 134, 241, 242
 core, 29, 30, 31, 78, 80, 81, 82, 83, 84, 97, 101, 152, 183, 184, 231, 236, 252
 CPU, 15, 20, 21, 78, 98, 172, 221, 222, 223, 224, 225
 crash, 10, 26, 27, 28, 29, 33, 76, 80, 81, 83, 97, 101, 180, 184, 228, 251
 crawl, 230
 createDocType, 134
 createProcessingRule, 134
 createTPA, 134
 CRM, 65
 daemon, 23, 52, 224
 dashboard, 162
 database, 11, 13, 18, 19, 22, 23, 24, 27, 34, 41, 43, 51, 53, 65, 69, 70, 88, 89, 90, 92, 93, 94, 95, 96, 111, 114, 116, 118, 120, 121, 122, 139, 140, 146, 149, 150, 152, 153, 155, 156, 157, 158, 164, 166, 168, 169, 180, 183, 184, 185, 187, 212, 216, 248, 249, 250
 DataDirect, 114, 168, 248, 249
 datasource, 93, 94
 DB2, 93, 94, 95
 Dblimits.cnf, 120
 dbx, 81, 224, 225
 DCC, 139
 DDL, 211
 deadlock, 22, 78
 decode, 33
 decrypt, 129, 243
 defragment, 72
 Deployer, 199, 200, 201
 DER, 128, 236, 238, 239, 240, 241, 242, 244
 Designer, 5, 68, 143, 144, 145, 146, 147, 148, 158, 162, 167, 168, 171, 173, 174, 177, 200
 Developer, 20, 42, 45, 68, 115, 123, 124, 137, 162, 167, 175, 202, 203, 204
 DHCP, 231
 diag.log, 61
 diagnostic_data.zip, 9
 dispatch.cnf, 14
 Dispatcher, 45
 display, 6, 12, 27, 49, 53, 54, 60, 140, 146, 152, 153, 156, 213, 219, 221, 225, 242
 DMZ, 48
 DNS, 49, 231
 document, 2, 5, 7, 17, 36, 42, 43, 44, 45, 59, 60, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72,

73, 74, 75, 76, 79, 117, 118, 122, 123, 125,
126, 127, 129, 134, 152, 155, 163, 164, 165,
166, 167, 168, 169, 218, 243

download, 82, 128

Dr. Watson, 30, 183, 184, 188, 251, 252

driver, 11, 19, 27, 88, 89, 90, 92, 93, 94, 95,
96, 114, 150, 168, 209, 248, 249

drwtsn32, 30, 80, 252

DTD, 187, 188

EAI, 102, 104

echo, 161

eclipse, 147, 148

EDI, 115, 123, 124, 125, 127

EDIINT, 119, 125, 126, 127, 128, 243

editor, 123, 169, 177

empower, 6, 201, 208, 231, 242

emulate, 242

encoding, 79, 140, 240, 241

encrypt, 128, 130, 243

envelope, 44, 79, 121, 122

ERP, 65

eService, 211

Excel, 213, 214

Exchange, 39

Fabric, 151

FileNotFoundException, 117

filter, 58, 95, 112, 198, 212

FIN, 132, 133, 134, 135, 137

Finalizer, 23, 24

firewall, 48, 49, 94, 233

Flow, 13, 18, 146, 203

folder, 31, 125, 146, 190, 201, 211, 216

foreign, 118, 212, 213

fragmentation, 72, 76

framework, 217

FTP, 83, 231

Garbage Collector, 227

gateway, 48, 49, 50, 127

gcore, 81

GCViewer, 230

gdb, 81

gencore, 81

getconfig, 157, 250

Glance, 222

Glue, 151, 237

grant, 119

graph, 230

grep, 48, 58, 220, 223, 248

gzip, 83

handshake, 234, 235, 242, 246

hang, 11, 20, 22, 23, 24, 25, 36, 66, 92, 96,
111, 116, 122, 167, 175, 206

hard disk, 7, 15

header, 130

heap, 17, 18, 19, 28, 76, 97, 98, 160, 173, 226,
227, 228, 229, 230

hexadecimal, 60, 222, 224, 225

HotSpot, 31, 224

HP, 29, 58, 76, 81, 222

HPjtune, 227, 230

HP-UX, 58, 76, 81, 222

hs_err_pid<pid>.log, 26, 29

httpd.conf, 181, 184

IBM, 28, 29, 34, 93, 110, 221, 227

IData, 134, 163

IDoc, 97, 98, 99

ifconfig, 231

IIS, 189

import, 86, 106, 118, 119, 132, 133, 134, 174,
201, 214, 215, 236, 241

inbound, 101, 125, 126, 234

index, 168, 180, 202

Inosrv.txt, 187

Insight, 190, 195, 197

inspectDB, 175

Installer, 102, 132, 162, 205, 206, 207, 208

InstanceId, 175

InstanceIteration, 175

Integration Server, 5, 6, 9, 10, 11, 12, 13,
 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26,
 27, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39,
 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
 52, 53, 54, 55, 56, 63, 64, 65, 66, 67, 68, 69,
 70, 73, 74, 75, 77, 78, 89, 90, 91, 92, 93, 94,
 95, 96, 97, 98, 100, 101, 102, 103, 104, 105,
 109, 112, 113, 114, 115, 116, 117, 118, 119,
 120, 121, 122, 124, 126, 128, 131, 132, 133,
 135, 136, 137, 138, 141, 142, 143, 144, 145,
 146, 149, 151, 152, 155, 156, 160, 162, 163,
 167, 168, 169, 170, 172, 173, 174, 175, 176,
 177, 199, 200, 201, 202, 204, 206, 219, 227,
 230, 232, 233, 234, 236, 241, 245, 246, 247,
 248, 249, 251
 iostat, 15
 IP, 41, 46, 110, 231, 232, 233
 ipconfig, 231, 233
 ISCoreAudit, 53, 69, 149
 ISInternal, 41, 51
 ISRuntimeException, 36
 issuer, 236, 237, 245
 J2EE, 31
 JAAS, 210
 jar, 40, 41, 47, 57, 73, 74, 75, 86, 87, 88, 103,
 104, 107, 113, 115, 151, 197, 205, 207, 209,
 248, 249
 Java, 10, 16, 17, 18, 21, 22, 23, 26, 27, 28, 29,
 31, 42, 56, 57, 73, 74, 75, 79, 85, 97, 98,
 103, 106, 116, 132, 143, 160, 186, 199, 204,
 206, 216, 220, 222, 223, 224, 226, 227, 229,
 251
 java.lang.SecurityException, 210
 java.net.SocketException, 215
 java.security.AccessControlException, 211
 java.sql.SQLException, 118, 150, 154
 javax.crypto.BadPaddingException, 247
 javax.security.auth.login.LoginException,
 210
 JBoss, 209
 JCE, 128
 JCo, 100
 JDBC, 11, 27, 29, 34, 41, 51, 53, 69, 88, 89, 90,
 92, 93, 94, 95, 96, 114, 117, 118, 122, 146,
 149, 150, 152, 154, 155, 173, 209, 226, 248,
 249
 JDBCpools, 9, 114
 JDK, 10, 104, 210, 229
 Jetty, 158
 JIT, 26, 34
 JKS, 236
 JMS, 47, 162, 170
 JNDI, 47, 210
 JNI, 29, 31
 jobs.cnf, 14
 join, 41, 163, 164, 165, 166, 171
 JRE, 207
 jRFC, 98, 100
 Jurisdiction, 128
 JVM, 10, 11, 12, 17, 18, 20, 21, 22, 23, 25, 26,
 27, 28, 29, 30, 31, 32, 34, 35, 36, 40, 46, 54,
 98, 101, 128, 160, 167, 197, 215, 219, 221,
 224, 226, 227, 228, 229, 230, 237, 245, 251
 jvm.dll, 104
 keepAlive, 49
 key, 7, 32, 55, 60, 118, 128, 130, 184, 191,
 212, 213, 232, 234, 235, 237, 239, 240, 241,
 242, 244, 246, 247, 251
 keystore, 236, 237
 keytool, 236, 237
 kill, 21, 81, 116, 183, 220, 224
 KPI, 140, 141
 LDAP, 59, 159, 216, 229
 leak, 18, 143, 230
 license, 2, 40, 41, 55, 59, 60, 151
 Linux, 18, 58, 80, 81, 182, 196, 206, 223, 229
 listener, 101, 112
 loop, 70, 167, 170, 202
 LWP, 222, 223
 manifest.v3, 34
 map, 43, 202, 203
 MarketSite, 121, 122

MDM, 209

MDN, 126, 243, 244

message, 12, 13, 21, 22, 34, 43, 47, 49, 51, 80, 88, 90, 97, 103, 108, 111, 112, 117, 118, 123, 125, 126, 128, 129, 130, 131, 134, 135, 139, 146, 154, 156, 158, 159, 163, 174, 176, 178, 186, 191, 196, 204, 206, 208, 211, 214, 215, 229, 234, 243, 246, 247

Metadata, 60, 211, 216

Microsoft, 15, 82, 143, 158, 229, 251

MIME, 126, 130, 243

model, 47, 154, 168, 171, 173, 199

Modeler, 131, 162, 171

Monitor, 5, 18, 19, 22, 23, 25, 57, 76, 80, 82, 131, 141, 145, 146, 149, 150, 151, 154, 155, 156, 163, 167, 168, 173, 174, 249

MQ, 109, 110, 111, 112, 113

multicast, 39, 40

MWS, 47, 114, 117, 143, 145, 153, 154, 155, 156, 157, 158, 159, 160, 161, 177, 178, 179, 235, 236, 237, 249, 250

My webMethods Server, 5, 6, 42, 56, 57, 61, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 114, 116, 117, 119, 124, 131, 137, 138, 140, 142, 143, 144, 145, 146, 147, 157, 158, 159, 160, 161, 173, 177, 178, 179, 199, 200, 201, 232, 235, 236, 237, 248, 249, 251

native, 26, 27, 28, 29, 31, 47, 57, 181, 222, 223, 224, 225, 226, 228, 229

netstat, 34, 46, 48, 49, 140, 232

network, 8, 37, 39, 40, 48, 49, 64, 76, 92, 94, 101, 110, 155, 231, 232, 233

node, 39, 40, 41, 115, 156, 157, 185, 190, 199, 200, 233

node.ndf, 199

nohup.out, 29, 220, 234

OCI, 93, 94, 95, 226

OneData, 209, 210, 211, 212, 214, 215, 216

OpenSSL, 236, 238, 239, 242, 244, 245, 246, 247

Optimize, 138, 140, 141, 142, 174

Oracle, 11, 28, 39, 54, 88, 89, 90, 93, 94, 95, 106, 149, 150, 214, 216, 248

OS390, 93

outbound, 22, 43, 101, 130, 207, 233, 234

package, 19, 34, 38, 55, 69, 70, 81, 83, 88, 92, 96, 102, 103, 104, 119, 121, 122, 123, 124, 125, 132, 134, 137, 144, 149, 155, 162, 168, 169, 174, 175, 199, 200, 204, 249

packet, 49, 233

partner, 118, 129, 130, 131, 134, 204, 236, 242

password, 103, 104, 106, 108, 177, 186, 206, 236, 237, 239, 249

PE, 162, 170

PE_NONTRANSACTIONAL_ALIAS, 162, 170

PEM, 236, 238, 240, 241, 242, 244

PERL, 239

PID, 220, 222, 223, 229

ping, 25, 45, 55, 94, 101, 199, 205, 231, 232

pipeline, 115, 167, 170, 203, 204

PKCS#12, 239, 241

plugmaker, 190, 191, 197

policy, 210, 211

polling, 37

port, 9, 10, 14, 25, 32, 39, 40, 46, 48, 49, 50, 73, 74, 75, 86, 87, 94, 96, 110, 116, 117, 140, 142, 145, 157, 192, 215, 232, 236, 237, 242, 245

portlet, 145

ports, 32, 34, 41, 49, 50, 232

process instance, 136, 146, 150, 151, 153, 154, 165, 169, 170, 171, 175

process model, 129, 131, 146, 151, 162, 163, 164, 165, 167, 170, 171, 173, 174, 199, 200, 201

ProcessAudit, 149, 154, 155, 168

ProcessExplorer, 225

progress, 35, 195

provider, 47, 245

proxy, 49, 207, 208, 233

prstat, 223

prt.cnf, 169, 173

ps, 15, 220, 223, 224

pstack, 183

publish, 36, 42, 44, 45, 46, 47, 72
 publish/subscribe, 42, 45, 46
 PublishAndWait, 44
 publisher, 63, 64, 68
 purge, 116
 putconfig, 157, 250
 QoS, 171, 174
 Quality Of Service, 174
 query, 55, 115, 139, 185, 249
 queryResults, 115, 116
 queue, 42, 43, 44, 59, 63, 64, 65, 66, 67, 68,
 69, 70, 71, 73, 74, 75, 77, 112, 113, 164,
 165, 166, 168
 RAC, 93, 94, 95
 RAM, 168, 172
 raw, 248, 249, 250
 readme, 6, 133, 176, 248
 real-time, 97
 ReceiverID, 127
 record, 29, 132, 161, 202
 recovery, 7
 registry, 30, 32, 135, 140, 151, 252
 reload, 121, 155
 repository, 13, 32, 92, 103, 104, 210, 216
 response, 20, 25, 44, 115, 116, 183, 205, 232,
 246
 RESTful, 216
 resubmission, 171
 resubmit, 152, 154, 171
 reverse, 48, 49, 50
 RFC, 97, 98, 99, 100, 101, 245
 RMI, 135, 215, 216
 RN, 130, 131
 RNO, 129, 130, 131
 RosettaNet, 129, 130, 131, 247
 round-robin, 50
 RSA, 234, 235, 240, 241, 243, 246
 rules, 118, 134, 201, 214
 S/MIME, 126, 243
 SAP, 27, 31, 97, 98, 99, 100, 101, 226
 SCC, 34, 88, 90, 106
 scheduler, 51, 52, 92, 116, 159
 schema, 41, 51, 53, 117, 134, 149, 152, 157,
 185, 187
 script, 47, 83, 139, 141, 196, 197, 208, 239,
 244, 248
 SDK, 191
 security, 48, 128, 186, 210, 215, 237, 245, 246
 segment, 85, 226
 sender, 42, 125, 127, 243, 244
 SenderID, 127
 sequence, 32, 33, 34, 35, 36, 44, 79, 135, 203,
 209, 212
 serial, 68
 server.cnf, 14, 19, 40, 43, 45, 52, 53, 54, 55,
 88, 118
 server_config, 43, 62, 64, 65, 66, 67, 68
 service, 6, 7, 8, 9, 10, 11, 12, 16, 19, 22, 27,
 32, 36, 37, 38, 42, 43, 44, 46, 51, 54, 55, 65,
 66, 69, 70, 77, 78, 90, 92, 93, 95, 96, 97, 99,
 101, 105, 109, 110, 111, 114, 116, 120, 121,
 127, 131, 133, 134, 137, 149, 150, 152, 153,
 154, 155, 156, 164, 166, 167, 170, 172, 175,
 199, 203, 204, 219
 ServiceNet, 151
 session, 9, 13, 39, 43, 53, 54, 55, 66, 67, 73,
 74, 75, 77, 115, 143, 232, 234, 246, 248
 shutdown, 21, 27, 36, 236
 Siebel, 102, 103, 104, 105, 106, 107, 108
 signal, 27, 76, 129, 188, 203
 signature, 129, 130, 243, 244, 247
 sl53_cj36.jar, 114, 248
 sleep, 34, 224
 SM59, 101
 SMH, 180, 181, 182, 184, 185
 smitty, 84
 SMP, 183
 SMTP, 177
 SOA, 180, 183, 190
 SOAP, 145, 167, 200

socket, 50
 Solaris, 58, 81, 183, 223
 spy, 249
 spy.log, 249
 SQL, 88, 90, 95, 139, 209, 249, 250
 SQLException, 88, 89, 90, 154, 211
 SSL, 60, 234, 236, 238, 242, 245, 246, 247
 stack, 13, 28, 29, 38, 41, 47, 81, 96, 117, 118, 156, 167, 174, 179, 183, 190, 191, 197, 208, 219, 221, 222, 229
 stacktrace, 186
 stateful, 54
 stateless, 54, 55
 static, 86, 106, 115
 stats, 9, 13, 227, 230
 stderr, 196
 stdout, 29
 storage, 17, 42, 43, 44, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 72, 73, 80, 83, 148
 string, 58, 79, 85, 163, 164, 165, 166, 190, 205
 struct, 79, 85
 subscribe, 6, 42, 47
 subscriber, 44, 68
 subscription, 44, 64, 163, 164, 165, 177
 Sun, 29, 31, 47, 226, 227, 229, 235
 SWIFT, 132, 133, 134, 135, 136, 137
 SWIFTNet, 132, 135, 136
 SYNONYM, 95
 sysadmin, 145, 161, 179, 249
 syslog.conf, 58
 Tamino, 183, 184, 185, 186, 187, 189
 Tangosol, 39, 40, 41
 tar, 83
 task, 51, 52, 54, 115, 177, 179, 201, 225
 TaskManager, 21
 telnet, 94, 140, 205, 232
 territory, 85, 86
 thread, 14, 19, 22, 23, 24, 25, 28, 29, 35, 36, 38, 42, 43, 46, 49, 51, 52, 65, 77, 78, 91, 92, 96, 99, 116, 167, 172, 173, 175, 179, 195, 204, 206, 219, 220, 221, 222, 223, 224, 225, 229
 TLS, 246
 TN, 9, 114, 116, 117, 120, 123, 127, 131, 165, 166, 167, 244
 TPA, 123, 131, 134
 Trading Networks, 5, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 134, 165, 167, 173, 201
 transaction, 38, 43, 47, 61, 62, 91, 101, 111, 125, 126, 127, 128
 transport, 36, 102, 121, 122, 127, 235
 trigger, 36, 42, 43, 44, 45, 46, 63, 64, 65, 66, 67, 68, 73, 74, 75, 77, 78, 90, 149, 150, 153, 167, 168, 169, 175, 177, 200, 212
 truncated, 81
 truss, 30, 35, 183
 trusted, 235, 243, 244, 245
 truststore, 236, 237, 245, 247
 tspace, 117, 118
 UDB, 93
 UDDI, 181
 ulimit, 81
 uname, 189
 unicode, 79, 85
 Unix, 15, 18, 21, 57, 59, 60, 76, 97, 180, 182, 183, 196, 229, 234, 235
 unlock, 65
 unresponsive, 20, 22, 23, 160, 179
 URL, 25, 93, 94, 122, 158, 190, 191, 192, 198, 209, 249
 user.dmp, 80
 userdump.exe, 82
 users.cnf, 14
 users.properties, 210
 verbose, 16, 17, 25, 29, 35, 83, 227, 242, 249, 250
 VeriSign, 244
 vmstat, 15
 volatile, 168, 171

Weblogic, 47

WebSphere, 109, 110, 111, 112, 113, 209

Windows, 15, 18, 21, 27, 30, 32, 57, 58, 59, 78,
80, 82, 97, 104, 106, 111, 178, 180, 181,
182, 183, 188, 189, 215, 219, 225, 229, 231,
232, 236, 251, 252

Wireshark, 233

WmDB, 19

Workflow, 216, 217

WSDL, 140, 145

WSP, 98

XML, 79, 127, 130, 183, 185, 200, 214, 215, 216