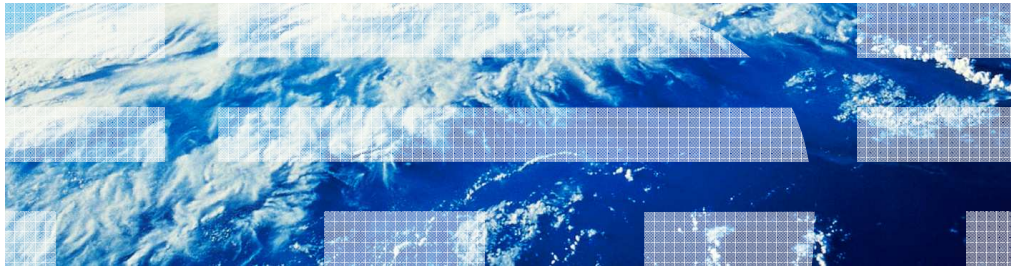


IBM WebSphere eXtreme Scale V8.6

Problem determination



© 2013 IBM Corporation

This presentation provides an overview of problem determination in IBM WebSphere® eXtreme Scale V8.6.

Table of contents

- Messages
- Log files
- Trace
- First Failure Data Capture (FFDC)

First, log messages generated by eXtreme Scale components will be described. Then, the presentation will cover normal output and error log files. Third, trace strings that provide more details on the workings of eXtreme Scale are discussed. And finally you learn about First Failure Data Capture capabilities and related files.

Message prefixes

- CWOBJ: eXtreme Scale core components
- CWPRJ: Entity projector
- CWWSM: HTTP Session Manager
- CWXQY: Query engine
- CWXSA: Extension point
- CWXSB: XsByteBuffer
- CWXSC: Console
- CWXSI: Command line
- CWXSR: Log analyzer
- SESN: HTTP Session Manager
- SSLC: SSL channel security
- TCPC: TCP channel
- WSBB: XsByteBuffer

© 2013 IBM Corporation

WebSphere eXtreme Scale produces messages with the prefixes shown here. The following ones are some of the most important.

“CWOBJ” messages are issued by the core ObjectGrid components. These messages describe the interactions between core components such as backing maps, grid server, and catalog servers.

“CWXQY” messages are issued by the query engine. These message can relate to both entity and object queries.

“CWPRJ” messages are issued by the Entity Manager projector components that convert data between entities and tuples, and vice versa.

“CWWSM” messages are issued by eXtreme Scale HTTP Session Manager.

Log files

- Server log files are provided for review of catalog and container server messages
 - SystemOut.log – look for start confirmation messages, peer mode messages, replication messages and so on
 - SystemErr.log – look for error messages
 - Trace file – exists if trace is enabled
- For stand-alone installations, the workingDirectory property in the server properties file is the root for the logs and optional trace file
 - Default is: <current_directory>/logs/<server_name>
- If eXtreme Scale is installed within a WebSphere Application Server environment (including the Liberty profile), see the WebSphere Application Server information center for more information about the location of server log files

Server log files allow you to review messages issued during server initialization, runtime, and termination. All messages are logged into SystemOut.log, which you can use to look for confirmation messages from various components within eXtreme Scale. The SystemErr.log provides a summary of only the error messages that have been reported.

For stand-alone installations, the location of the log directory is controlled by the “workingDirectory” property in the server properties file. If a working directory is not specified, the runtime will create a default *logs* directory under the “current” directory. The server's logs are in a subdirectory with the same name as the server.

Optionally a trace file can exist within the log directory, if trace is enabled. You can direct the trace file to another location using the “traceFile” parameter in the server properties file.

If WebSphere eXtreme Scale is installed within a WebSphere Application Server environment, including the Liberty profile, then review the WebSphere Application Server information center if you have questions about specifying log locations.

Methods to set trace

- Supply trace arguments in command invocation
- Server properties file
- Using the xscmd utility
- Collect trace on a stand-alone Java client
- WebSphere Application Server administrative console
- Programmatically

There are several ways to control tracing. One option is to specify the trace settings when you start the servers. To do this, use the **-traceSpec** argument on the server start command. A second option is to specify the trace settings in the server properties file. A third option is to use the xscmd utility to configure the trace settings and to turn tracing on and off dynamically.

If eXtreme Scale V8.6 is installed in a WebSphere Application Server environment, then a fourth option is to use the administrative console to define the trace strings and trace file size. The process for defining these trace settings for eXtreme Scale is exactly the same as for other components within a WebSphere Application Server process.

Finally, a fifth option is to programmatically start and stop tracing.

Trace arguments on server start command

- Setting trace on a stand-alone server at start time
 - traceSpec – to specify trace specifications
 - traceFile – to specify the path and trace file name
- Example:

```
startXsServer.sh catalogServer -traceSpec ObjectGrid*=all -traceFile /home/user1/trace.log
```

Here are more details about using the trace arguments on the server start command to configure the trace settings when you start a stand-alone server. Specify the **-traceSpec** argument to control which components are traced. Specify the **-traceFile** argument to control where the trace log file is written. If you do not specify the **-traceFile** option, the trace file is written to the same directory as the SystemOut.log and SystemErr.log files. An example is shown here, where both the **-traceSpec** and **-traceFile** arguments are specified on the startXsServer command.

Configuring trace using a server properties file

- Create the properties file with trace properties

```
Example: server.properties  
traceSpec=ObjectGrid*=all  
traceFile=C:/temp/trace.log  
systemStreamToFileEnabled=true
```

© 2013 IBM Corporation

Another way to configure trace settings is to use a properties file. First, create a properties file named **server.properties**. In the file, specify the trace settings by using the **traceSpec** property. Optionally you can specify where you want the trace file to be written using the **traceFile** property. The **systemStreamToFileEnabled** property controls whether the **System.out** messages are written to **SystemOut.log** and **System.err** messages are written to **SystemErr.log**.

How eXtreme Scale finds the properties file

- Reference the properties file in the command `-serverProps`
- Auto-discovery of well-named properties file anywhere in the classpath
 - `objectGridServer.properties`
- System property
 - `objectgrid.server.props=<properties file name>`
- Programmatically
 - `ServerFactory.getServerProperties()`
- Sample properties files:
 - `<extreme_scale_installation_root>/ObjectGrid/properties`
 - `<websphere_installation_root>/optionalLibraries/ObjectGrid/properties`

You can choose how to have the system recognize your properties file. You can modify the script or command you use to start the server, specifying the `serverProps` argument to reference the properties file you created. Or you can use the “well-named properties file”, `objectGridServer.properties`, which the system can locate if it is within the classpath. Finally you can define a system property, `objectgrid.server.props`, to point to your properties file.

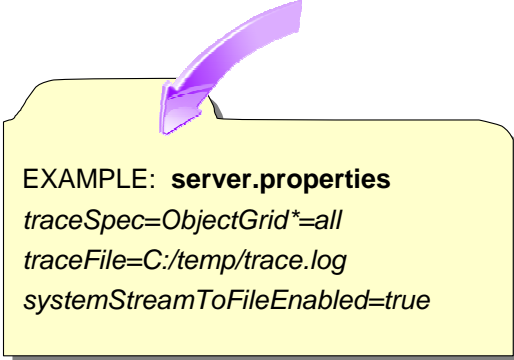
The eXtreme Scale information center also discusses a programmatic way to locate a properties file using the `ServerFactory` class.

Sample properties files are provided within the stand-alone eXtreme Scale installation in the `<extreme_scale_installation_root>/ObjectGrid/properties` directory. For an eXtreme Scale installation in a WebSphere Application Server environment, the sample properties files are located in `<websphere_installation_root>/optionalLibraries/ObjectGrid/properties`.

Properties file in a command invocation

- Example of specifying a properties file within a command invocation

```
startXsServer.sh server1 <server start arguments> -serverProps  
../properties/server.properties -jvmArgs -cp <classpath> <args to pass to JVM>
```



EXAMPLE: **server.properties**
traceSpec=ObjectGrid=all*
traceFile=C:/temp/trace.log
systemStreamToFileEnabled=true

Here is an example of specifying a properties file within a command invocation. Notice the JVM arguments must be the last arguments within the command invocation.

Collect trace using the xscmd utility

- Examples

```
xscmd -c setTraceSpec -spec ObjectGridReplication=all
```

```
xscmd -c setCatTraceSpec -spec ObjectGridPlacement=all
```

To collect trace with the xscmd utility, use the setTraceSpec command. This allows you to change the trace specification and turn trace on or off dynamically. You can collect trace on all container servers. You can use the setCatTraceSpec command to collect trace on all catalog servers. You can filter the servers using the s or sl parameter.

Collect trace on a stand-alone Java client

- You can start trace collection on a stand-alone client by adding system properties to the startup script for the client application. In this example, trace settings are specified for the `com.ibm.samples.MyClientProgram` application:

```
java -DtraceSettingsFile=MyTraceSettings.properties  
-Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager  
-Djava.util.logging.configureByServer=true  
-cp <path containing MyTraceSettings.properties>  
com.ibm.samples.MyClientProgram
```

EXAMPLE: **MyTraceSettings.properties**

`traceFileName=c:/MyTraceFile.log`

`ObjectGridRouting=all`

© 2013 IBM Corporation

When stand-alone client applications that access eXtreme Scale have problems, it might be useful to enable tracing for the application. Enabling trace for client programs will cause the eXtreme Scale classes used by the applications to generate trace information.

To enable trace for the eXtreme Scale classes in a client application, add the system properties to the startup script or command of the client application. The location of the output and the trace specification are shown in the example. The trace settings file specified, `MyTraceSettings.properties` in the example, must be a properties file located in a directory that is contained in the class path of the application client or stand-alone process.

Additional ways to set trace

- Use WebSphere Application Server administrative console
 - For eXtreme Scale installations in WebSphere Application Server environments
 - Use same way as other WebSphere components
 - See the WebSphere Application Server information center for information about working with trace
- Use programmatic invocation during runtime
 - Set trace during runtime using ObjectGridManager
 - See the WebSphere eXtreme Scale information center for more information about setting trace programmatically

If eXtreme Scale is installed within a WebSphere Application Server environment, use the WebSphere Application Server administrative console to set tracing for eXtreme Scale and to control the size and location of log files. See the WebSphere Application Server information center for details about how to perform this task.

You can programmatically set tracing within your application at runtime. This is covered in more detail in the WebSphere eXtreme Scale information center.

Trace string format

- `<ObjectGridComponent>=<level>`
- You can concatenate trace strings; use ":" as separator

```
ObjectCatalogServer=all:ObjectGridPlacement=all
```

- Use the * symbol to specify a wild card

```
ObjectGrid*=all
```

The trace string specification has two basic parts: the component to be traced and the trace level. You can specify more than one trace string specification by separating the strings with a colon. You can use the asterisk as a wild card to enable more than one trace component with the same prefix in one specification.

Trace components (1 of 4)

- eXtreme Scale ObjectGrid components for tracing

Component	Description
ObjectGrid	<i>General core cache engine</i>
ObjectGridCacheInvalidator	<i>Near-cache invalidation</i>
ObjectGridCatalogServer	<i>General catalog service and server runtime</i>
ObjectGridChannel	<i>Catalog cluster communication</i>
ObjectGridClientInfo	<i>DB2® client information.</i>
ObjectGridClientInfoUser	<i>DB2 user information.</i>
ObjectGridConfig	<i>XML configuration file parsing</i>
ObjectGridContinuousQuery	<i>Continuous query</i>
ObjectGridCORBA	<i>ORB transport communication</i>
ObjectGridDataGrid	<i>AgentManager API</i>
ObjectGridDynaCache	<i>eXtreme Scale dynamic cache provider</i>
ObjectGridEntityManager	<i>EntityManager API</i>

© 2013 IBM Corporation

The table on this slide and the next few pages show the eXtreme Scale components that can be traced and a brief description of each component.

Trace components (2 of 4)

- eXtreme Scale ObjectGrid components for tracing

Component	Description
ObjectGridEvictors	<i>Built-in evictors</i>
ObjectGridJPA	<i>eXtreme Scale-specific Java Persistence API; Java Persistence API loaders</i>
ObjectGridJPACache	<i>Java Persistence API L2 cache plug-ins</i>
ObjectGridLocking	<i>cache entry lock manager</i>
ObjectGridLogHandler	<i>Remote logging information</i>
ObjectGridMBean	<i>Management beans</i>
ObjectGridMonitor	<i>Historical statistics monitoring infrastructure</i>
ObjectGridNative	<i>eXtreme Scale native code</i>
ObjectGridOSGi	<i>eXtreme Scale OSGi integration</i>
ObjectGridPlacement	<i>Catalog and container server shard placement service</i>
ObjectGridPubSub	<i>Near-cache invalidation and continuous query communication</i>
ObjectGridQuery	<i>Object Query API and EntityManager Query API</i>

© 2013 IBM Corporation

This slide shows more of the components and associated descriptions for tracing.

Trace components (3 of 4)

- eXtreme Scale ObjectGrid components for tracing

Component	Description
ObjectGridReplication	<i>Replication service</i>
ObjectGridRest	<i>REST gateway</i>
ObjectGridRouting	<i>Client/Server routing details</i>
ObjectGridSecurity	<i>eXtreme Scale security</i>
ObjectGridSerializer	<i>Data serializer plug-in</i>
ObjectGridSpring	<i>eXtreme Scale Spring integration</i>
ObjectGridStats	<i>eXtreme Scale statistics</i>
ObjectGridTransactionManager	<i>eXtreme Scale transaction manager</i>
ObjectGridWriteBehind	<i>eXtreme Scale write behind</i>
ObjectGridXA	<i>Multi-partition transaction</i>
ObjectGridXDF	<i>eXtreme Data Format</i>
ObjectGridXIO	<i>eXtremeIO transport communication</i>

© 2013 IBM Corporation

This slide shows more of the components and associated descriptions for tracing.

Trace components (4 of 4)

- eXtreme Scale ObjectGrid components for tracing

Component	Description
ObjectGridXIOChannel	<i>eXtremeIO transport communication</i>
ObjectGridXM	<i>eXtremeMemory</i>
ObjectGridXMEviction	<i>eXtremeMemory eviction</i>
Projector	<i>EntityManager API engine</i>
QueryEngine	<i>Query engine for the Object Query API and EntityManager Query API</i>
QueryEnginePlan	<i>Query plan</i>
TCPChannel	<i>eXtremeIO TCP/IP channel</i>
WXSRevision	<i>Revision control for replication</i>
XsByteBuffer	<i>eXtreme Scale byte buffer</i>

This slide shows the remainder of the components and associated descriptions for tracing.

Trace levels

- Common trace levels
 - all – all trace records
 - debug – debug trace records
 - entryExit – entry and exit trace records
 - event – event trace records
- <ObjectGridComponent>=<level>
- Example

```
ObjectGridPlacement=entryExit
```

- If unsure of level, use “all”

The trace levels provide you control over the amount of detail you want traced. If you do not know what trace level to specify when providing diagnostic traces to IBM support, use “all” to ensure that the necessary data is included in the trace log.

First Failure Data Capture directory and files

This section will provide an overview of the First Failure Data Capture directory and files.

First Failure Data Capture files

- First Failure Data Capture files are for IBM support to assist in debug
- Will be requested by IBM support for problem diagnosis
- Appear in a directory labeled ffdc (in the logs directory) and contain files that resemble:
 - server2_exception.log
 - server2_6580658_13.03.15_11.14.14.0425346424866757635681.txt
- First Failure Data Capture files are written for servers only

WebSphere eXtreme Scale provides “First Failure Data Capture” data, also called FFDC data. This data is primarily intended to be used by IBM support.

For stand-alone installations, the location of the log directory is controlled by the “workingDirectory” property in the server properties file. If a working directory is not specified, the runtime will create a default *logs* directory under the “current” directory. The First Failure Data Capture logs are in an ffdc subdirectory in the logs directory. The ffdc directory will contain an exception summary log for each server (shown in the example as “server2_exception.log”). It will also contain one or more reports for each server with file names composed of the server name and a time stamp. The contents of the ffdc directory should be sent to IBM support along with the SystemOut.log, SystemErr.log, and trace.log files.

First Failure Data Capture data is written for server processes only and not produced for client processes.

Section

Summary

The next slide provides a brief recap of this presentation.

Summary

- Messages
- Log files
- Trace
- First Failure Data Capture (FFDC)

In Summary, WebSphere eXtreme Scale uses specific message prefixes to indicate which product component issued the message. You can access the log files to review catalog and container server messages, which can help determine if the problem that is occurring is during server start or during server runtime. The trace facility allows you to turn on trace, specify particular components for tracing, and specify particular trace levels to control the amount of detail in the trace log. First Failure Data Capture data is available to provide to IBM support when you report problems.

References

- IBM Elastic Caching Community
<http://www.ibm.com/developerworks/connect/caching>
- IBM WebSphere eXtreme Scale Version 8.6 Information Center
<http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6>
- IBM WebSphere Application Server Version 8.5 Information Center
<http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5>

See these references for additional information about **WebSphere eXtreme Scale**



Feedback

Your feedback is valuable

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

1. Did you find this module useful?
2. Did it help you solve a problem or answer a question?
3. Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_XS86_ProblemDetermination.ppt

This module is also available in PDF format at: [../XS86_ProblemDetermination.pdf](..../XS86_ProblemDetermination.pdf)

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



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DB2, System p, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

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

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2013. All rights reserved.