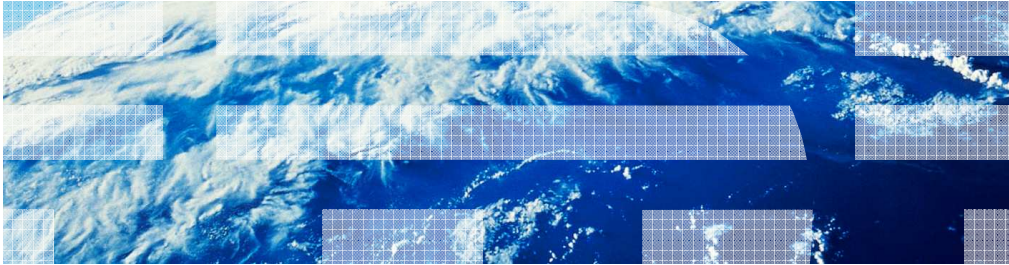


# IBM WebSphere Extreme Scale V8.6 WebSphere eXtreme Scale Client for .NET

## Configuration



© 2013 IBM Corporation

This presentation describes how to configure the WebSphere® eXtreme Scale Client for .NET that is included in IBM WebSphere eXtreme Scale V8.6

## .NET client configuration

- There are three possible ways to configure the .NET client.
  1. Through the client properties file.
  2. Property values received from the server properties.
  3. Programmatically overriding received server set properties through .NET client public interface.
- Client properties file
  - Has properties for general client configuration, client security, credential authentication, transport layer security and SSL configurations.
  - The default client properties file is Client.Net.properties and is located under `<install location>\eXtreme Scale Client for .NET\config`
  - Is read during the client's Connect() call to the grid.
  - A custom client properties file can be specified by the clients using a fully qualified path name to the client properties file during Connect() call.
  - When an empty client properties file is specified during a Connect() call, all the client properties are initialized to use the appropriate default values.
  - Once the client establishes connection with the server, any changes to the client properties file can take effect only by restarting the client's connection to the grid.

There are three ways to configure the .NET client.

The first is through the client properties file that is specified during the initial Connect() call to the grid.

The second is through values received from the server. These properties are received by the client when it makes its connection with the server. They are TransactionTimeout, LockTimeout, and TimeToLive, which are set in the objectgrid.xml configuration file while starting the server.

Third, you can programmatically override the server set properties using the .NET client API.

The client properties file has property values to configure client security, credential authentication, transport layer security, and SSL configuration parameters. Configuration through the client properties file occurs during the .NET client's Connect() call to the grid. If the properties file is not specified during the Connect() call, the client properties are initialized with the values defined in the installed Client.Net.properties file located under the config directory of the installation location. After the client calls the Connect() method, any changes to the client properties file can take effect only by restarting the client's connection to the grid.

## Client properties file

- Properties that can be set through the client properties file:

Property name	Possible values	Default value	Notes
preferZones	N/A	N/A	Must be CSV e.g.: ZoneA,ZoneB,ZoneC
requestRetryTimeout	Any value >=0	-1	When requestRetryTimeout value is set to -1, the timeout is governed by transaction timeout
shuffleBootstrapAddresses	True, False	true	Specifies if the catalog service grid addresses should be randomized when used by a client while bootstrapping to the grid.
securityEnabled	True, False	false	If security is to be enabled on the client side, the server must turn on the security as well.
credentialAuthentication	Never, Supported, Required	Supported	Credential authentication requirement.
authenticationRetryCount	Any value >=0	0	Value 0 indicates that retry will not happen
credentialGeneratorAssembly	N/A	N/A	Must be a valid C# dll name with version, properties and other properties included.
credentialGeneratorClass	N/A	N/A	Must be a valid C# class name and is used to generate credential object for the client.
credentialGeneratorProps	N/A	N/A	String containing username and password
transportType	SSL-Supported, SSL-Required, TCP/IP	SSL-Supported	Set the transport type.
commonName	N/A	N/A	Not used
Protocol	Ssl2,Ssl3,Tls,Default	N/A	Specify the protocol to use.
publicKeyFile	N/A	N/A	Path to server's public key file.

3

Configuration

© 2013 IBM Corporation

This table briefly summarizes the property values that can be configured using the client properties file. All their possible values and default values are summarized.

## Server properties and overriding

- Server properties and programmatically overriding through .NET public interface
  - The .NET client receives a set of properties from the server when the client establishes its connection with the server.
  - Properties received on the client from the server configuration can be get or set (overridden) using the .NET client interfaces.
  - A call to `IGridMapPessimisticTx.ResetToDefaults()` resets all overridable properties to the values when the grid was obtained by .NET client.
  - A detailed list of overridable properties are listed in the following slides.

The .NET client also receives several server-side XML configuration property values that can be overridden programmatically using the .NET client APIs. All the programmatically overridden values on the .NET client can be reset back to the initial values of the grid by calling the `ResetToDefaults()` method.

## Overridable server properties (1 of 3)

- Properties that can be overridden from .NET public interfaces:

Property (get: read only, set: overridable)	.NET client interface	Possible values	Default value	Readable /settable	Notes
TtlEvictorType(get)	IGridMapPessimisticAuto	NONE, LAST_ACCESS_TIME, LAST_UPDATE_TIME	NONE	Read only	.NET client can only retrieve the value. Set through the server config (objectgrid.xml)
	IGridMapPessimisticTx				
TimeToLive(get,set)	IGridMapPessimisticAuto	Any value >=0	0 (never expires)	Overridable	Can be set through the objectgrid.xml and can be get/set (overrides) programmatically by the .NET client.
	IGridMapPessimisticTx				
LockTimeout(get,set)	IGridMapPessimisticAuto	Any value >=0	15 seconds	Overridable	Can be set through the objectgrid.xml and can be get/set (overrides) programmatically by the .NET client.
	IGridMapPessimisticTx				
numPartitions(get)	IPartitionManager	N/A	N/A	Read only	Retrieves the number of partitions set in objectgrid.xml

The following three slides summarize the attributes that can be set or overridden by the .NET client API.

## Overridable server properties (2 of 3)

Property {get: read only, set: overridable}	Interface	Possible values	Default value	ReadOnly/o verridable	Notes
TransactionTimeout {get,set}	IGrid	Any value >=0	-1 (indicates value not set)	Overridable	Retrieves the number of partitions set in objectgrid.xml. <b>Value set to TimeSpan.Zero indicates unlimited timeout.</b>
	IGridTransaction	Any value >=0	10 minutes		
IsSecurityEnabled {get}	IGrid	True, false	False	Read only	Retrieves the value set in Client.Net.properties.
Name {get}		N/A	N/A	Read only	Retrieves the name of the grid.

The transaction timeout can be set either on the grid using the IGrid interface or on the transaction itself using the IGridTransaction interface.

## Overridable server properties (3 of 3)

Property {get: read only, set: overridable}	Interface	Possible values	Default value	Readable/ settable	Notes
Active{get}	IGridTransaction	True, False	N/A	Read only	To determine if a transaction is active or not.
MarkedRollbackOnly {get}		True, False	N/A	Read only	Returns true if a transaction is marked as rollback only.
Name {get}	IGridMap	N/A	N/A	Read only	Retrieves the name of the Map
Grid {get}		N/A	N/A	Read only	Retrieves the Grid instance
PartitionManager {get}		N/A	N/A	Read only	Retrieves the map's partition manager

The properties listed here can be read, but not overridden, by the .NET client API.

## Troubleshooting

### ▪ Troubleshooting

- The client application receives a thrown exception for most of the cases concerning failures with .NET client properties configuration. This includes misspelled or wrong client properties file, and illegal values for properties.
- When the specified client properties file is not found as specified during the Connect() call, the SystemTrc.log file will result in the following entry.

```
Current Date: 2012-11-27 11:45:57.8035 Base Directory: C:\Program Files (x86)\IBM\WebSphere\eXtreme Scale Client for .NET\sample\SimpleClient\bin\Debug
***** End Display Current Environment *****
[11/27/12 11:45:57.753522-05:00] 00000001 Client.ApiImpl.GridManager
D Connect: Error parsing properties, System.IO.FileNotFoundException 'Could not find file 'C:\Program Files (x86)\IBM\WebSphere\eXtreme Scale Client for .NET\config\nochere.properties'.'
```

- When a null string is specified in place of client properties file during a Connect() call, all the properties in client properties file are initialized to use default values.
- Inappropriate values specified for a given property in client properties file will result in an error message in the SystemOut.log and SystemErr.log. Also in SystemTrc.log when trace is enabled. Sample:

```
***** End Display Current Environment *****
[11/27/12 13:05:04.566024-05:00] 00000001 Client.Properties.ClientProperties E CW0BJ0008E: The value wrongvalue provided for the property credentialAuthentication is not valid.
```

Any failure with .NET client properties file configuration, including bad property values, or bad or misspelled client properties file location, results in an exception thrown from the Connect() method. When a custom path to a property file is specified, and if the file is not present in the path, a message will be logged in the SystemTrc.log file. Similarly, when an illegal value is specified on a property, the result is an error entry logged in the SystemTrc.log when trace is enabled. When a null client properties file is specified during the Connect() call, then the client property values are initialized using the default values that are defined in the installed Client.Net.properties file located under the config directory of the installation location.



## 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\\_Net\\_Configuration.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_XS86_Net_Configuration.ppt)

This module is also available in PDF format at: [./XS86\\_Net\\_Configuration.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, 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.