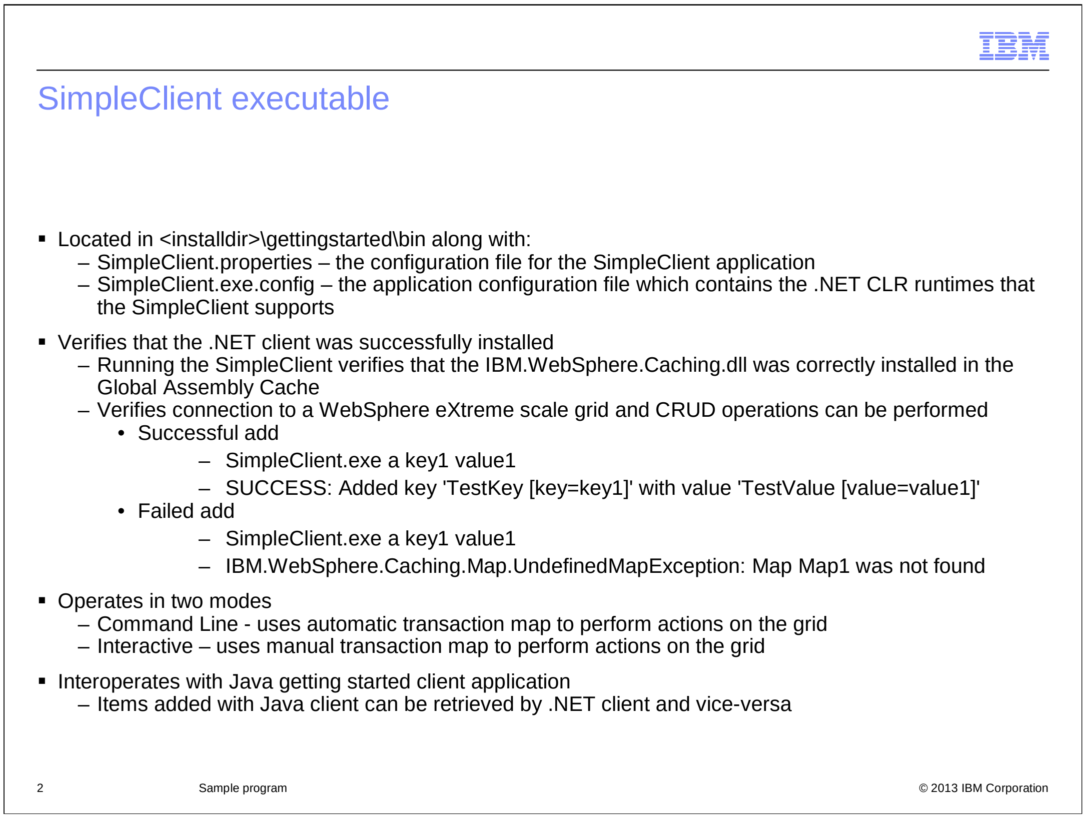
IBM WebSphere Extreme Scale V8.6 WebSphere eXtreme Scale Client for .NET

Sample program

This presentation describes how to use the sample program to perform basic operations on the grid using the WebSphere® eXtreme Scale Client for .NET.



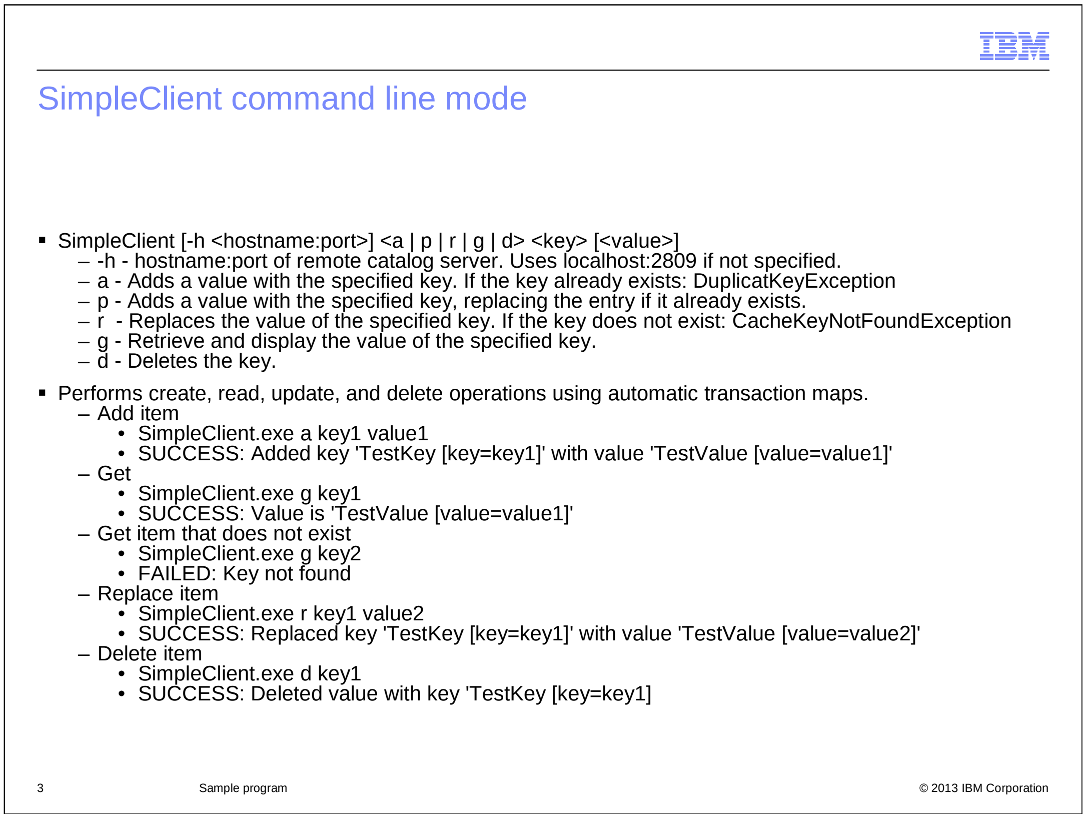
SimpleClient executable

- Located in <installdir>\gettingstarted\bin along with:
  - SimpleClient.properties – the configuration file for the SimpleClient application
  - SimpleClient.exe.config – the application configuration file which contains the .NET CLR runtimes that the SimpleClient supports
- Verifies that the .NET client was successfully installed
  - Running the SimpleClient verifies that the IBM.WebSphere.Caching.dll was correctly installed in the Global Assembly Cache
  - Verifies connection to a WebSphere eXtreme scale grid and CRUD operations can be performed
    - Successful add
      - SimpleClient.exe a key1 value1
      - SUCCESS: Added key 'TestKey [key=key1]' with value 'TestValue [value=value1]'
    - Failed add
      - SimpleClient.exe a key1 value1
      - IBM.WebSphere.Caching.Map.UndefinedMapException: Map Map1 was not found
- Operates in two modes
  - Command Line - uses automatic transaction map to perform actions on the grid
  - Interactive – uses manual transaction map to perform actions on the grid
- Interoperates with Java getting started client application
  - Items added with Java client can be retrieved by .NET client and vice-versa

2 Sample program © 2013 IBM Corporation

The sample executable file is located in the gettingstarted\bin directory in the installation directory. The directory includes a SimpleClient.properties file that specifies the client configuration parameters for the sample application and a SimpleClient.exe.config file that contains the .NET CLR runtime information that the sample application supports. Running the sample program verifies that the .NET client was successfully installed and that IBM.WebSphere.Caching.dll was successfully loaded in the Global Assembly Cache.

The sample program operates in two modes: the command line mode and the interactive mode. Command line mode uses the automatic transaction map to perform the operations on the grid whereas the Interactive mode uses the manual transaction map. This program also interoperates with the Java client’s Getting Started application such that items added with the Java client can be retrieved by the .NET client and vice versa.

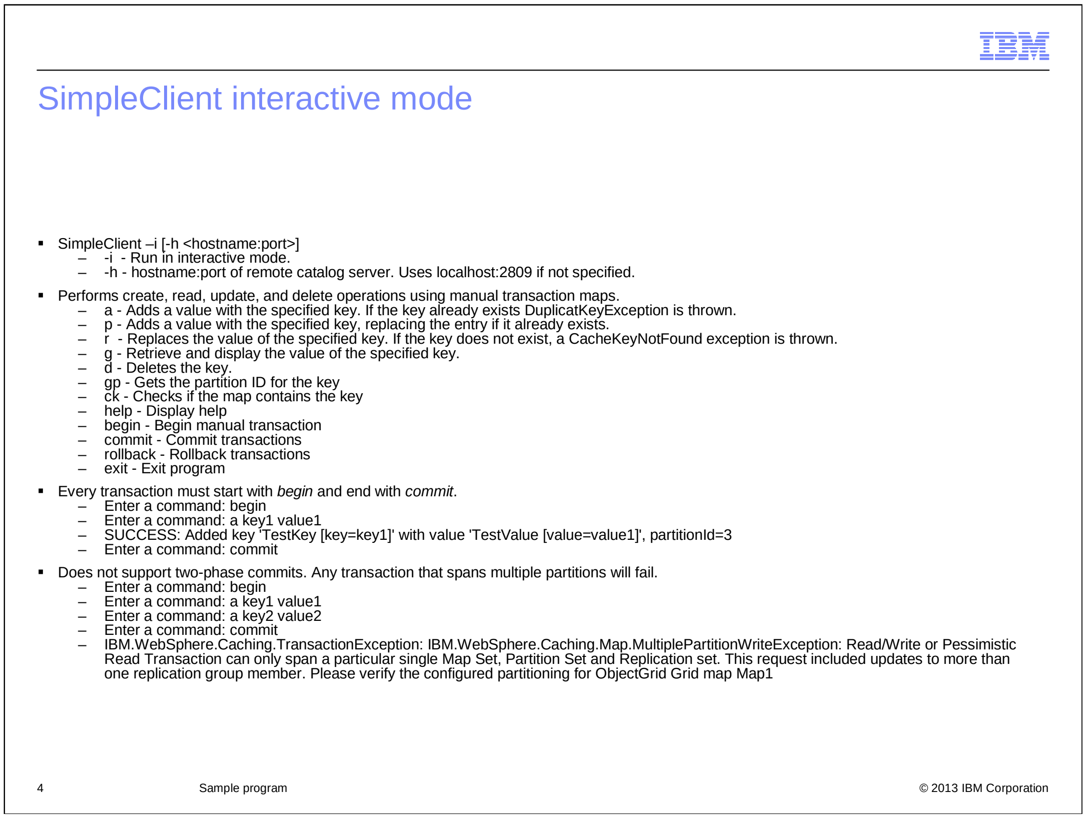


SimpleClient command line mode

- SimpleClient [-h <hostname:port>] <a | p | r | g | d> <key> [<value>]
  - -h - hostname:port of remote catalog server. Uses localhost:2809 if not specified.
  - a - Adds a value with the specified key. If the key already exists: DuplicatKeyException
  - p - Adds a value with the specified key, replacing the entry if it already exists.
  - r - Replaces the value of the specified key. If the key does not exist: CacheKeyNotFoundException
  - g - Retrieve and display the value of the specified key.
  - d - Deletes the key.
- Performs create, read, update, and delete operations using automatic transaction maps.
  - Add item
    - SimpleClient.exe a key1 value1
    - SUCCESS: Added key 'TestKey [key=key1]' with value 'TestValue [value=value1]'
  - Get
    - SimpleClient.exe g key1
    - SUCCESS: Value is 'TestValue [value=value1]'
  - Get item that does not exist
    - SimpleClient.exe g key2
    - FAILED: Key not found
  - Replace item
    - SimpleClient.exe r key1 value2
    - SUCCESS: Replaced key 'TestKey [key=key1]' with value 'TestValue [value=value2]'
  - Delete item
    - SimpleClient.exe d key1
    - SUCCESS: Deleted value with key 'TestKey [key=key1]

This slide describes the parameters that can be used to run the sample application in command line mode. The '**–h'** parameter is used to specify the hostname and port number that must be used to connect to the remote catalog server. If this option is not specified while executing the sample application, the default localhost:2809 will be used. The parameter '**a**' is used to add a key value pair to the grid. If the specified key is already present, then a DuplicateKeyException error is returned. The parameter **'p'** is also used to add a key value pair to the grid. The difference between parameter a and p is that the parameter p replaces the entry with the specified one if the key is already found in the grid. The parameter '**r**' is used to replace the value specified by the key. If the specified key is not found, then a CacheKeyNotFoundException error is returned. The parameter '**g**' retrieves and displays the value specified by the key. And the parameter '**d**' deletes the key value pair entry from the grid.

Some sample uses of these parameters are shown here.

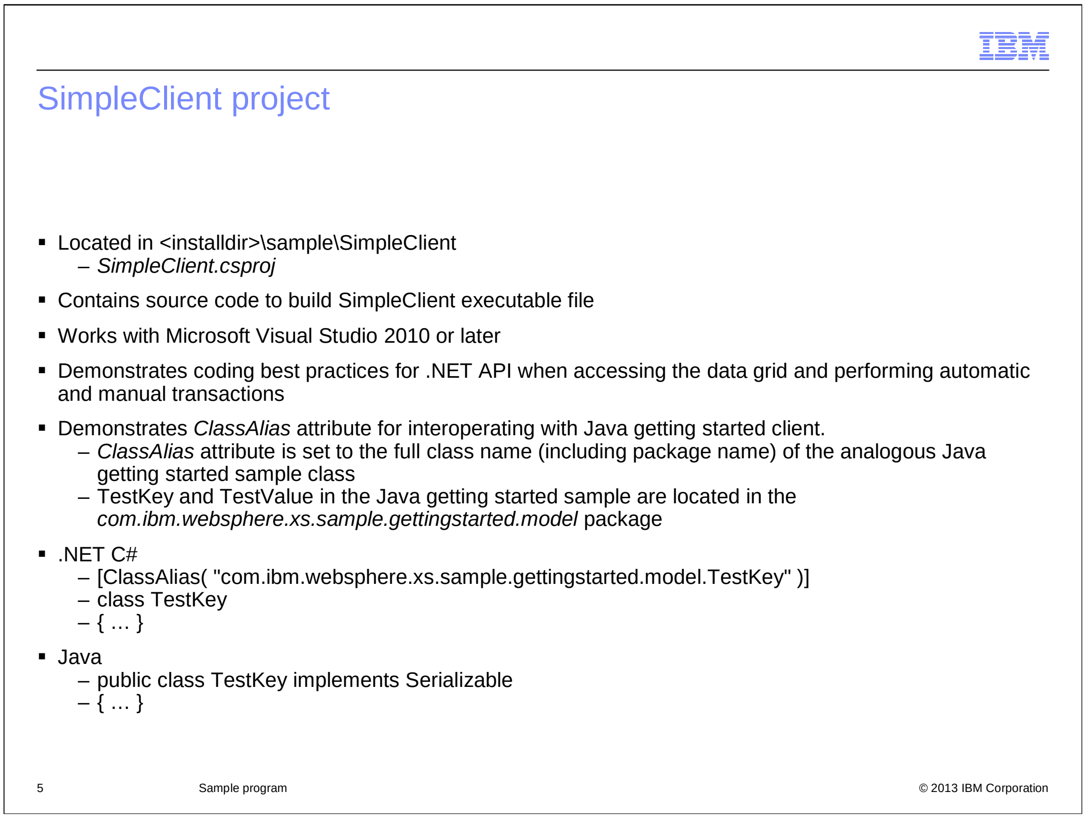


SimpleClient interactive mode

- SimpleClient –i [-h <hostname:port>]
  - -i - Run in interactive mode.
  - -h - hostname:port of remote catalog server. Uses localhost:2809 if not specified.
- Performs create, read, update, and delete operations using manual transaction maps.
  - a - Adds a value with the specified key. If the key already exists DuplicatKeyException is thrown.
  - p - Adds a value with the specified key, replacing the entry if it already exists.
  - r - Replaces the value of the specified key. If the key does not exist, a CacheKeyNotFound exception is thrown.
  - g - Retrieve and display the value of the specified key.
  - d - Deletes the key.
  - gp - Gets the partition ID for the key
  - ck - Checks if the map contains the key
  - help - Display help
  - begin - Begin manual transaction
  - commit - Commit transactions
  - rollback - Rollback transactions
  - exit - Exit program
- Every transaction must start with *begin* and end with *commit*.
  - Enter a command: begin
  - Enter a command: a key1 value1
  - SUCCESS: Added key 'TestKey [key=key1]' with value 'TestValue [value=value1]', partitionId=3
  - Enter a command: commit
- Does not support two-phase commits. Any transaction that spans multiple partitions will fail.
  - Enter a command: begin
  - Enter a command: a key1 value1
  - Enter a command: a key2 value2
  - Enter a command: commit
  - IBM.WebSphere.Caching.TransactionException: IBM.WebSphere.Caching.Map.MultiplePartitionWriteException: Read/Write or Pessimistic Read Transaction can only span a particular single Map Set, Partition Set and Replication set. This request included updates to more than one replication group member. Please verify the configured partitioning for ObjectGrid Grid map Map1

In the interactive mode, a manual transaction map is used. Every transaction must be manually started using a “begin” statement and committed using “commit” statement. The parameter ‘**-i**’ is used to start the program in interactive mode. A sample is shown here. The parameter options a,p,r,g, and d work the same as in command line mode. The only difference is that every single operation has to be manually started and committed.

The .NET client does not support two-phase commits. Any transaction that spans multiple partitions will fail; therefore, if you try to perform more than one operation between a “begin” and a “commit” statement using keys that do not map to the same partition, a MultiplePartitionWriteException error is produced.

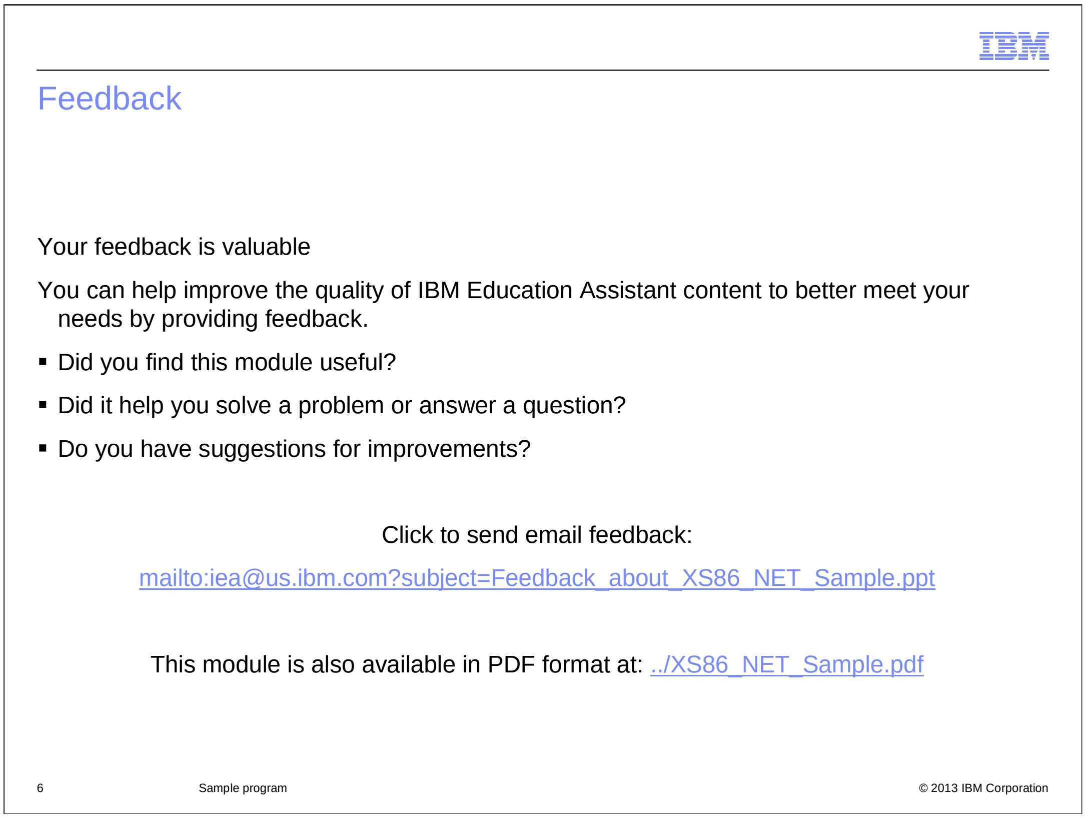


SimpleClient project

- Located in <installdir>\sample\SimpleClient
  - *SimpleClient.csproj*
- Contains source code to build SimpleClient executable file
- Works with Microsoft Visual Studio 2010 or later
- Demonstrates coding best practices for .NET API when accessing the data grid and performing automatic and manual transactions
- Demonstrates *ClassAlias* attribute for interoperating with Java getting started client.
  - *ClassAlias* attribute is set to the full class name (including package name) of the analogous Java getting started sample class
  - TestKey and TestValue in the Java getting started sample are located in the *com.ibm.websphere.xs.sample.gettingstarted.model* package
- .NET C#
  - [ClassAlias("com.ibm.websphere.xs.sample.gettingstarted.model.TestKey")]
  - class TestKey
  - { … }
- Java
  - public class TestKey implements Serializable
  - { … }

The getting started SimpleClient project source is located in the installation directory under sample\SimpleClient. This sample project works with Microsoft Visual Studio 2010 or later versions. Open the project by Double-clicking on SimpleClient.proj solution file which will automatically load the project in Microsoft Visual Studio. To build the project, click Build->Build Solution. The project references the IBM.WebSphere.Caching.dll located in the \bin directory in install location. If this project is moved to a different location, it may not build properly. The sample program also demonstrates coding best practices for using .NET client API to access the data grid and to perform automatic and manual transactions.

The ClassAlias attribute can be used to share data grid data between a similar Java class and a .NET class to enable interoperability with the Java client’s getting started sample client. The ClassAlias attribute must be set to the full class name, including the package name of the analogous Java client’s sample.



Feedback

Your feedback is valuable

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

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

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_XS86_NET_Sample.ppt

This module is also available in PDF format at: ../XS86_NET_Sample.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" at http://www.ibm.com/legal/copytrade.shtml

Microsoft, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

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.