NCache 4.6 Administrator’s Guide
Table of Contents

1. Preface.......................................................................................................................... 6

2. Configuring Caches ........................................................................................................ 2
   2.1. Creating Caches ........................................................................................................ 2
       2.1.1. Creating Local Cache ...................................................................................... 2
       2.1.2. Creating Clustered Cache ................................................................................ 6
   2.2. Adding Server Node in Cache .................................................................................. 13
   2.3. Removing Server Node from Cache ........................................................................ 15
   2.4. Configuring Basic Cache Settings .......................................................................... 17
       2.4.1. Cache Size ........................................................................................................ 17
       2.4.2. Cache Isolation Level (InProc/OutProc) ......................................................... 17
   2.5. Configuring Expiration ............................................................................................ 19
   2.6. Cache Data Format .................................................................................................. 20
       2.6.1. Configuring Data Format ................................................................................ 21
   2.7. Configuring Cache Cluster Settings ........................................................................ 23
       2.7.1. TCP Port .......................................................................................................... 23
       2.7.2. Port Range ....................................................................................................... 23
       2.7.3. Operation Timeout .......................................................................................... 24
       2.7.4. Statistic Replication Interval .......................................................................... 24
       2.7.5. Connection Retries ......................................................................................... 24
       2.7.6. Retry Interval .................................................................................................. 25
   2.8. Configuring Query Indexes ...................................................................................... 26

2.9. Registering Classes for Compact Serialization ......................................................... 31
   2.9.1. Registering Non-Generic Classes ....................................................................... 31
   2.9.2. Registering Generic Classes ............................................................................... 36
   2.9.3. Registering Generic Classes using Generic Type Handler ................................. 41

2.10. Registering Classes for Portable Data Sharing ......................................................... 43
   2.10.1. Import/Export Data Sharing Configuration ....................................................... 50

2.11. Configuring Data Source Providers .......................................................................... 53
2.11.1. Configuring Read-Through Provider .......................................................... 53
2.11.2. Configuring Write-Through Provider ....................................................... 58
2.11.3. Configure Write-Through Providers for Write-Behind Operations .......... 62
2.12. Configuring Cache Start-Up Loader ............................................................. 64
2.13. Configuring Cache Level Events ................................................................. 68
2.14. Configuring Eviction Policy ........................................................................ 68
2.15. Configuring Compression ........................................................................... 70
2.16. Configuring Email Notifications/Alerts ....................................................... 70
2.17. Binding Cluster and Client/Server with Dedicated IP for Performance .......... 72
  2.17.1. Binding Cluster with a Dedicated IP ......................................................... 72
  2.17.2. Binding Client/Server with a Dedicated IP ............................................. 74
2.18. Configuring MapReduce ............................................................................. 76
2.19. Configuring Heartbeat ................................................................................. 77
2.20. Add Existing Clustered Caches ................................................................... 78
3. Management Operations .................................................................................. 80
  3.1. Start Cache ................................................................................................... 80
  3.1.1. Starting Cache on All Server Nodes ....................................................... 80
  3.1.2. Starting Cache on a Particular Server Node ......................................... 81
  3.2. Stop Cache .................................................................................................. 83
  3.2.1. Stop Cache on all Server Nodes ............................................................ 83
  3.2.2. Stopping Cache on a Particular Server Node ........................................ 84
  3.3. Manage Cache Service on a Server Node ................................................... 86
  3.3.1. Start Cache Service on a Server Node .................................................. 86
  3.3.2. Stop Cache Service on a Server Node .................................................. 86
  3.3.3. Restart Cache Service on a Server Node ............................................. 87
  3.4. Apply Configuration .................................................................................... 89
  3.5. Hot Apply Configuration ............................................................................ 90
  3.6. Reset Configuration .................................................................................... 91
  3.7. Data Load Balancing ................................................................................... 92
4. Configuring Cache Clients .............................................................................. 94
  4.1. Adding Client Node ..................................................................................... 94
NCache 4.6 Administrator's Guide

6. Configuring Bridge for WAN Replication ................................................................. 121

6.1. Overview of Bridge ................................................................................................. 121

6.1.1. Active-Passive ...................................................................................................... 121

6.1.2. Active-Active ....................................................................................................... 122

6.2. Creating Clustered Bridge ..................................................................................... 123

6.3. Adding Clustered Caches to the Bridge ................................................................. 128

6.4. Configuring Bridge Settings .................................................................................. 131

6.4.1. Configuring Bridge Queue Settings .................................................................. 131

6.4.2. Configuring Bridge Connection Settings ......................................................... 132

6.4.3. Enable Queue Optimization ............................................................................... 133

6.4.4. Configuring Bulk Size for Operation Replications ........................................... 134

6.4.5. Performance Counters ....................................................................................... 135

6.5. Configuring Conflict Resolver ............................................................................... 136

6.6. Changing Cache Synchronization Modes (Active/Passive) .................................. 141

6.6.1. Making a Passive Cache as Active Cache ......................................................... 141

6.6.2. Making an Active Cache as Passive Cache ......................................................... 141

6.7. Configuring Master Cache ..................................................................................... 142

6.8. Force State Transfer .............................................................................................. 143

6.9. Connect/Disconnect a Cache .................................................................................. 144
## Monitoring

### 6. Bridge Management
- **6.11. Remove a Cache from the Bridge**: 146
- **6.11.2. Apply Configuration**: 147
- **6.11.4. Remove a Bridge Node**: 149
- **6.11.5. Start Bridge**: 150
- **6.11.6. Stop Bridge**: 152
- **6.11.7. Remove Bridge**: 153
- **6.11.8. Start Bridge Service**: 154
- **6.11.9. Stop Bridge Service**: 155
- **6.11.10. Restart Bridge Service**: 156

### 7. Monitoring
- **7.1. Cache Counters**: 157
  - **7.1.1. Performance Counters**: 157
  - **7.1.2. Debugging Counters**: 159
- **7.2. Monitoring Caches Using NCache Manager**: 160
  - **7.2.1. Changing the Management Port of a Cache Node**: 160
  - **7.2.2. Configuring Counters to Display**: 162
  - **7.2.3. Cluster Health**: 164
  - **7.2.5. Monitoring Clustered Caches Using NCache Email Alerts**: 166
- **7.2. Monitoring Caches Using NCache Monitor**: 167
  - **7.2.1. Monitoring Cache Server Counters Using Built-In Dashboard**: 167
  - **7.2.2. Monitoring Cache Client Counters Using Built-In Client Dashboard**: 170
  - **7.2.3. Monitoring Clustered Cache Counters using Custom Dashboard**: 171
  - **7.2.4. Monitoring PerfMon Counters Using Custom Dashboard**: 171
  - **7.2.5. Monitoring Cache Event Logs Using Custom Dashboard**: 173
  - **7.2.6. Monitoring Client Process of a Clustered Cache Using Custom Dashboard**: 176
  - **7.2.7. ASP.NET Session Monitoring in Custom Dashboard**: 179
- **7.3. Monitoring Bridge Using NCache Manager**: 181
  - **7.3.1. Configuring Bridge Counters to Display**: 181
7.3.2. Browse Bridge Statistics ................................................................. 183
7.4. Logging ................................................................................................................ 184
  7.4.1. NCache Log Viewer ................................................................. 184
  7.4.2. NCache Monitor Logging ......................................................... 192
  7.4.3. Windows Event Logging ............................................................ 193
  7.4.4. Performance Counters Logging .............................................. 193
7.5. Monitoring NCache Using Windows PerfmonTool ........................................ 195
  7.5.1. Monitoring Cache Server Counters Using PerfmonTool .......... 195
  7.5.2. Monitoring Cache Client Counters Using Perfmon Tool ............ 200
  7.5.3. Monitoring Bridge Counters Using Perfmon Tool ................. 205
8. IPV6 Support ........................................................................................................... 209
1. Preface

Welcome to NCache version 4.6! NCache is a flexible and feature rich 100% .NET caching solution that provides high performance and scalability to handle any transaction load. NCache features and APIs are designed to cache data for applications of any size – from small to enterprise-wide global installations.

This NCache documentation contains samples, tutorials, and tools designed to help you quickly incorporate NCache into your applications. You can search for topics in the extensive class library references, overviews and step-by-step illustrations.

Support

NCache provides various sources of technical support. Please refer to http://www.alachisoft.com/support.html to select a support resource you find suitable for your issue.

To request additional features in the future, or if you notice any discrepancy regarding this document, please drop an email to support@alachisoft.com.

Document Conventions

The following conventions in text have been used throughout this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bold</td>
<td>Specifies terms of importance for the reader.</td>
</tr>
<tr>
<td>Monospace</td>
<td>Specifies any packages and interfaces to be added into the application.</td>
</tr>
<tr>
<td>courier</td>
<td>Specifies file, class, interface names.</td>
</tr>
<tr>
<td>monospace</td>
<td>Specifies command line snippets.</td>
</tr>
<tr>
<td></td>
<td>Specifies additional and vital information for the user.</td>
</tr>
</tbody>
</table>
2. Configuring Caches

2.1. Creating Caches

This section provides information about creation and configuration of caches.

2.1.1. Creating Local Cache

Local cache can be created in the following ways:

**Using NCache Manager**

- Select **File -> New -> Cache** option.

- In **New Cache Wizard**, specify the server IP address and click **Next**.
• Specify the **name** of the cache in the text box, select the **Isolation level**, **Data format** and click **Next**.

Cache with the specified name should not exist on the server; otherwise you get an error message that this cache already exists on this server.

• Change the cache size if required and click **Next**.
In Advanced options dialog, you can configure the following settings:

- You can change the **eviction policy**. The eviction policy will be configured and can be turned on.
- You can also change the **eviction percentage**; items will be evicted from the cache using this percentage value if eviction is enabled.
- You can change the **clean interval value**. Default is 15 seconds.
- This cache can be started automatically upon the completion of this process if you have checked the **Start this cache on Finish** checkbox on this dialog.
- You can make the cache to start automatically after service restart by checking the checkbox **Auto start this cache on service startup**.
Click **Finish**. Created cache will be appeared in **Cache Explorer** under **Local Caches** node hierarchy.

**Using Command Line Tool**

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
- Use `createcache.exe`, provide the required information:
  
  `createcache.exe demoLocalCache /s 20.200.21.95 /S 1024`
  
- Press **ENTER**, it will create demoLocalCache of local topology of size 1024MB on node 20.200.21.95.
2.1.2. Creating Clustered Cache

Clustered cache can be created in following ways:

**Using NCache Manager**

- Select File -> New -> Clustered Cache option.

- **New Clustered cache** wizard will open. Specify the **name** of the cache, select **Data Format** and click **Next**.
Cache with the specified name should not exist on the server; otherwise you get an error message that this cache already exists on this server.

- In Select caching topology dialog, select the required cache topology. You can also change the Operation timeout (sec.) and Statistic replication interval (sec.) under the Advanced cluster options.
- Specify the desired information and click Next.
- If you selected the Partition Replica Cache topology, you have to select a Replication Strategy, either Asynchronous or Synchronous.
- Click Next.
- Specify the node IP address in text box or select from the **Network Neighborhood** and click on the > button to add it in the **Selected Nodes** list.

- Click **Next** once all the required nodes are added.
- Change the **Cluster Port** if required.
- Click **Next**.
• In the next dialog, you can change the **cache size** if required.

• Click **Next**.

![New Clustered Cache Wizard](image)

**In Advanced options** dialog, you can configure the following settings:

- You can change the **eviction policy**. The eviction policy will be configured and can be turned on.
- You can also change the **eviction percentage**; items will be evicted from the cache using this percentage value if eviction is enabled.
- You can change the **clean interval value**. Default is 15 seconds.
- This cache can be started automatically upon the completion of this process if you have checked the **Start this cache on Finish** checkbox on this dialog.
- You can make the cache to start automatically after service restart by checking the checkbox **Auto start this cache on service startup**.
Click Finish. Created cache will appear in Cache Explorer under Clustered Caches section.

Using Command Line Tool

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
• Use `createcache.exe` tool, provide required information:

```
createcache.exe ClusteredCache /s 20.200.21.95 /S 1024 /t replicated /C 7802
```

• Press **ENTER**, it will create ClusteredCache of replicated topology on server 20.200.21.51 with size 1024MB on port 7802.
2.2. Adding Server Node in Cache

A Server Node can be added in cluster any time. After adding a server node in a cluster, you need to start that node so that it can join the cluster.

Using NCache Manager

- Right click on a cache name in Cache Explorer and select Add Server Node option.
- Add Node Wizard will open. Specify the server IP which you want to add in text box and click Finish.
- Added node will appear in **Server Nodes** list under this cache name.
Using Command Line tool

- Open command prompt.
- Go the NCache tools directory: `cd %NCHOME%/bin/tools`.
- Use `addnode.exe`, provide required information:
  
  ```
  ```

- Press ENTER, it will add 20.200.22.107 node in demoLocalCache.

2.3. Removing Server Node from Cache

A server node from the cache can be removed in the following way:

Using NCache Manager

- Expand the cache and in Server Nodes section, right click on the node which you want to remove from the cache.
• Select the Remove option.
• NCache Manager will ask for confirmation, click Yes. Server nodes list will be refreshed and node will be removed from the cache.

Using Command Line Tool
• Open command prompt.
• Go to the NCache tools directory: cd %NCHOME%/bin/tools
• Use removenode.exe, provide required information:

  removenode.exe demoClusteredCache /s 20.200.21.107

• Press ENTER, it will remove node 20.200.22.107 from demoClusteredCache.
2.4. Configuring Basic Cache Settings

NCache gives you the full control on cache configurations so that you can create and configure a cache according to your requirements. When you click on the cache name in **Cache Explorer**, you can see different tabs at the right side.

In this section you will learn how to configure or customize a few basic settings using these tabs.

2.4.1. Cache Size

Through this configuration option, you can specify an upper limit on the size of the cache in MB. The number of items in the cache are limited by maximum size of the cache and amount of free system memory (virtual or physical). If some eviction policy is specified, whenever total data size in cache reaches maximum size or when less than 10% of memory is left in the system, items are removed from the cache as per eviction policy. Maximum size of a cache can be specified by one of following two ways.

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings
- Go to the **Main** tab.
- Change the **Cache Max Size** as per your needs.

- Right click the cache in **Cache Explorer** and select **Apply Configuration** option.

2.4.2. Cache Isolation Level (InProc/OutProc)

If your application is running on a server node (the node that is participating in the cluster), you can access the cache either as InProc or OutProc. For InProc, cache lives inside your application process and performs all the clustering operations from there. For OutProc, you have to start the cache independently and then connect to it. Both access modes have their own pros and cons.

InProc mode in a replicated cache can give you really fast GET operations since you are accessing all the data from within your own memory space. However, since your application and the cache are sharing the same memory, you may face memory size limitation (in terms of how much you can cache).

Similarly, OutProc has benefit that multiple applications on the same machine can share a common cache. Additionally, since the cache lives in its own memory, you have more memory available to you. But, there is an overhead of transferring data between your application process and the NCache process.
OutProc requires that your .NET Assemblies (the ones that contain the objects you are going to add into NCache) are either registered in GAC or are located at $INSTALL/NCache/bin/service folder. Otherwise, NCache will not find your object definitions which it requires for .NET Serialization. And, this will result in NCache throwing an exception when you do Add, Insert, Get, or Remove operations.

If you are running NCache in a clustered configuration, you must ensure this on all machines in the cluster. By default, all caches are OutProc but isolation level can be changed for caches of local topology. It can be done in the following way.

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to **Main** tab.
- Change the cache **Isolation Level** as per your requirement.

> Specify application isolation level. A high isolation level implies that the cache runs in Cache service's process whereas a Low level implies that Cache is InProc to the applications. An isolated cache can be shared between applications on the same node. Moreover, an isolated cache's lifetime is explicitly controlled by using applications like NCache Manager, Cmd Tools or APIs.

- Right click on cache name in **Cache Explorer** and select **Apply Configuration** option.
2.5. Configuring Expiration

Clean interval is the periodic interval after which the expired items are removed from the cache. By default, expiration Clean-Up interval is 15 seconds. However, it is changeable by the following way:

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to **Policies** tab.
- Change the **Clean Interval** as per your needs.
- You can choose any default expiration policy, and change its invalidation time accordingly.

<table>
<thead>
<tr>
<th>Expiration Policy</th>
<th>Clean Interval (sec):</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

**Note:** If Default Expiration is enabled and cache API without expiration is used then item is added with Absolute Default Expiration.

- Right click on cache name in **Cache Explorer** and select **Apply Configuration** option.
2.6. Cache Data Format

When the client sends an object to the cache, the user object, e.g. Product class, is stored in either of the two data formats in NCache – Binary or Object. The data format is an encoding of the data for storage in the cache, and the format you choose depends on the type of operations you expect to perform over the cache.

Note that when data travels across the network, it has to be serialized regardless of the data format being used. Serialization saves the state of the object in a linear stream of bytes for socket communication. The same serialized object is then reconstructed back to its original form when needed by the client, granted that the assemblies for the custom object (e.g. Product.dll) have been provided.

While serialization is particularly useful in aiding transportation of objects over a network, it comes with a resource overhead of the CPU and IO devices being utilized in the serialization and deserialization process. In case compression or encryption is configured, additional overhead may incur.

We will take an example of the Product class object in the following discussion:

**Binary Format**

In the binary format, the user objects (e.g. Product object) are serialized from the client end and stored on the server side in the same form. Every time an item is requested from the server, the client receives the binary form of the item which is then de-serialized to Product object locally. The size of an item in binary format is significantly smaller compared to the Object format. Furthermore, serialization/deserialization takes place on the client side only, and just once either while fetching or adding the data – de-serialized while fetching and serialized while adding. This saves cost of serialization/de-serialization, which is noticeable especially in cases where sizeable data is added or fetched from the cache.

**Object Format**

In case of Object format, the user object is serialized before being sent to the server. The server receives the serialized object de-serializes it back to the Product form and stores it in that format. When the client requests for the object, the server takes the de-serialized object from the cache, serializes it, and then sends it back to the client. The client then de-serializes the binary object to its original form (in this case, Product). This additional serialization and deserialization on the server side leads to twice the cost of the process. Moreover, it should be ensured that the custom object libraries have been deployed on the server side as well, as deserialization will be taking place on the server.

**Uses**

**Binary Format**

Binary Format is beneficial if most of your processing is on the client side, and the operations performed are like add, update, fetch, and remove from the cache. For example, a serialized Product object is fetched from the cache to display its contents. Using binary format, the item will only need to be de-serialized once it reaches the client, thus, keeping the processing cost to a minimum. In case Object format is being used, the server will serialize the Product object and send it to the client which will then de-serialize it. This increases the overall overhead and cost of serialization and de-serialization.

Similarly, if an object is being added to the cache using Binary format, it will be serialized before it is sent over the network and it will be stored as it is. In case Object format is used the item will first be serialized on the client side and then de-serialized on the server side before being stored.
Object Format

However, if local processing is required on the server side, without fetching and adding the items back from client like Entry Processors, Aggregator and MapReduce, Object format is required.

For example, you regularly execute an Entry Processor over your cache items. The execution will be performed on the Product objects. If the user object has been stored on the server side as a Product object, it will directly execute the code over the item. If the object has been stored in binary format, then the item will have to be de-serialized first to execute the entry processor.

Moreover, Object formats are valuable in cases where a data source is involved, like loading data from data source on startup (CacheLoader), saving changes made to cache items to the data source (Write-Through) or reading data from the data source (Read-Through). This is because the data is stored in the data source in the object format, no matter what the format is in the cache. If the object is stored in the Object format, there will be no need for additional deserialization apart from the socket serialization for data transfer.

2.6.1. Configuring Data Format

By default, NCache uses the Binary Format. However, in case of Local cache and Client cache, if it is InProc it will take Object format as default, and if OutProc, it will take Binary format as default. You can configure the format for OutProc cache according to your requirements.

Using NCache Manager

You can configure the data format through the Manager in two ways:

1. During cache creation:
2. Through the **Main** tab:
   - Click a cache in **Cache Explorer** to open cache settings.
   - Go to the **Main** tab.
   - Change the cache **Data Format** as per your requirement.

   Specify how items should be stored in Cache. 'Binary' format is a default option where items are serialized from client end and are stored on the server side as is. Each time the item is requested, it is received in binary form which is then deserialized at client end. If data format is 'Object', items are serialized at client end before they are sent to server. Server receives the serialized object and deserializes it before it is stored in the cache. When this item is requested by client, server takes the deserialized object from cache store, serializes it and then sends it to client. Client then receives the binary object and deserializes it. 'Object' format is recommended only for data analytics e.g. Entry Processor etc.

   ![Data Format: Binary](data_format.png)

   - To de-serialize data in the cache, class assemblies must be deployed on the server (in case of Object format).
   - Click on the **Deploy Task Libraries** button at the bottom left of the **MapReduce** tab.
   - Browse for the required assemblies, select them and then click **Open**.

   ![Deploy Task Libraries](deploy_task.png)

   - Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.
2.7. Configuring Cache Cluster Settings

NCache uses TCP ports for communication. Default TCP port is 8250. These ports can be changed, if required, by changing the values of the registry key (TCP.PORT) written upon the installation of NCache.

This key can be found under the following path:

`HKEY_LOCAL_MACHINE/SOFTWARE/ALACHISOFT/NCACHE`

Please note that NCache Manager application assumes the default port. Therefore, upon changing the values of this port, Local Cache/Cluster cannot be administered through NCache Manager application. However, NCache command line utilities and tools provide the option to address the Local Caches/Clusters on other ports.

2.7.1. TCP Port

TCP port is also assigned by default. It must be unique 4-digit number and can be changed in the following way:

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings
- Go to Cluster Settings tab.
- Change the **Cluster Port**, if required.

- Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.

2.7.2. Port Range

Port Range can be changed in the following way:

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to Cluster Settings tab.
- Change the **Port Range**.

- Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.
2.7.3. Operation Timeout

Operation Timeout is the maximum time in seconds during which a clustered operation must be completed, otherwise the operation will fail. Default Timeout value is 60 seconds and cannot be less than that. Operation Timeout can be changed in the following way:

- **Using NCache Manager**
  - Click on the cache name in **Cache Explorer** to open cache settings.
  - Go to Cluster Settings tab.
  - Change the Operation timeout (sec.).

  ![Advanced cluster options]
  
  **Operation timeout (sec.):** 60

  ![Advanced cluster options]
  
  **Statistical replication interval (sec.):** 500

  - Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.

2.7.4. Statistic Replication Interval

This option is available for Clustered Cache only. Statistics replication interval is the time interval in seconds after which a node sends its statistics to all other nodes in a cluster.

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to the Cluster Settings tab.
- Set the Statistic replication interval (sec.) as per your requirement.

  ![Advanced cluster options]
  
  **Operation timeout (sec.):** 60
  
  **Statistical replication interval (sec.):** 500

  - Right click the cache in **Cache Explorer** and select **Apply Configuration** option.

2.7.5. Connection Retries

Nodes participating in a cluster are connected with each other through TCP connections which may break up due to network failure. In that case, a client node tries to reestablish the connection. It goes through all the servers in the list once and if none is available, then it sleeps for Retry Interval seconds and then scans the list again. This is done Connection Retries times. In other words, the value of Connection Retries specifies how many times the node should try to restore the connection. Connection Retries can be changed in the following way:
Using NCache Manager

- Click on the cache name in Cache Explorer to open cache settings.
- Go to Cluster Settings tab.
- Set the **Connection Retries** as per your requirement.

  Nodes participating in a cluster are connected with each other through TCP connections, which may breakup due to any network failure. In that case, a node tries to reestablish the connection. Connection Retries specify how many times a node should try to restore the connection.

  ![Connection Retries](image)

- Right click on the cache name in Cache Explorer and select **Apply Configuration** option.

2.7.6. **Retry Interval**

Retry Interval specifies the time between two successive connection retries. Retry interval can be changed in following ways:

**Using NCache Manager**

- Click the cache name in Cache Explorer to open cache settings.
- Go to Cluster Settings tab.
- Change the **Retry Interval (sec.)** as required.

  ![Retry Interval](image)

- Right click on the cache name in Cache Explorer and select **Apply Configuration** option.
2.8. Configuring Query Indexes

Indexes are used in querying for efficiency, and are based on the type of a class. The user may further specify the attributes of the class on which indexes are required. Query Indexes can be configured in the following ways:

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to Query Indexes tab.
- Click **Add**.

Indexes are used in queries for efficiency. Indexes are based on the type of a class. The user may further specify the attributes of the class on which indexes are required. Note that currently you cannot run any query that has non-indexed attributes in its **where** clause.

**Selected Classes and their attributes:**

- **Select Query Indices** wizard will open. Click **Browse** to browse for required assembly. Select the assembly.

- Assembly and all its classes will be listed in **Selected Assemblies** section.
Select the required classes by checking the check box and click **Add Class** button.
List of selected classes along with its attributes will be listed down in **List of selected Classes** section. Select the attributes you want to add indexes for and click **OK**.
NCache Manager displays all private members and properties of selected class. Select either private members or properties.

- **Select Query Indexes** wizard will close and selected classes and their attributes will appear in 'Selected Classes and their attributes' section of 'Query Indexes' tab.
Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.

### Using Command Line Tool

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
- Use `addqueryindex.exe`, provide required information:

  ```
  addqueryindex.exe demoLocalCache /a C:\temp\Data.dll /c Data.Product /L ProductID$ProductName$Category$UnitsAvailable
  addqueryindex.exe demoLocalCache /a C:\temp\Data.dll /c Data.Customer /L CustomerID$CustomerName
  ```
- Press **ENTER**, it will add ProductID, ProductName, Category, Supplier and UnitsAvailable attributes from the class Data.Product and Data.Customer of Data.dll in query indexes for demoLocalCache.
2.9. **Registering Classes for Compact Serialization**

Each custom class must be serialized before adding any of its objects into cache. NCache compact serialization is used to dynamically serialize classes (generic or non-generic) on the fly.

2.9.1. **Registering Non-Generic Classes**

You can use either of the two ways described below to register your classes for compact serialization.

**Using NCache Manager**

- Click on the cache name in the **Cache Explorer**. It will open cache configuration settings in right pane.
- Go to the **Compact Serialization** tab.

*Compact Serialization Framework serializes/deserializes the objects efficiently than the native binary serialization in terms of time and space. This improves the performance of the application.*

- Click the **Add Types** button, a new **Select Compact Classes** dialog box will open up.
Click on Browse. It will open a windows Open dialog. Using this dialog you can select .exe, .dll or .jar file.

Select your required file (.exe, .dll or .jar) and click on Open. It lists classes from the selected assembly in Loaded Class list box.

Select desired classes, which you want to register with NCache for compact serialization.

Click Add Class.
Compact serialization works on attribute level. In the lower pane, attributes of all your selected classes get listed; from here you can select attributes, which you do not want to be compact serialized.
Click **OK** to add classes. It lists fully qualified names of all selected classes in **Compact Serialization** tab.
Right click on the cache name in Cache Explorer and select Apply Configuration option.

If cache is already running, then you can apply configuration by clicking on the Hot Apply Configuration in the right click menu.

Using Command Line Tool

- Open command prompt
- Go to the NCache tools directory like: `cd %NHOME%/bin/tools`
- Using `addcompacttype.exe`, provide the required information like:

  ```
  addcompacttype.exe /id demoLocalCache /a C:\Data.dll
  
  addcompacttype.exe /id demoLocalCache /a C:\Data.dll
  /cNCache.Sample.Data.Comact.Manager /al hireDate
  ```

- Press ENTER. It defines a compact type for demoLocalCache. You can define further types one by one repeating the step for other types.
2.9.2. Registering Generic Classes

NCache also supports generic classes to be compact serialized. Generic classes are configured in the same way as other classes. In addition we also need to specify `<T>` types for each Generic class.

- Click on the cache name in Cache Explorer to open cache settings.
- Go to the Compact Serialization tab.
- Click on the Add Type button. Select Compact Classes dialog appears.

Click on the Show Default Generics button. It lists default generic types.

Select the desired type and click Add Class.

If you want to select your own custom generic class, then click on the Browse button in the same dialog box and select your assembly.
- It adds selected classes in the lower pane. Click **OK** to add classes for compact serialization.
- All of the selected classes are now listed at **Compact Serialization** tab in NCache Manager.
• Now you need to specify the classes, which you expect to be assigned to the selected generic classes. When you select any of the added generic class, the Add<T> button becomes enabled.

• Click Add<T>.

• Select Compact Classes dialog appears. Click Browse to select assembly.

• All classes will be loaded in Loaded Classes list box.

• Select your class and click Add Class.

If you want to add primitive type as <T>, then click on Add Primitive Types button.
• Click **OK**. It adds generic `<T>` type under the selected generic class.
This way you can specify as many `<T>` types for a generic class as needed.

### 2.9.3. Registering Generic Classes using Generic Type Handler

NCache also gives you the facility to configure all required generic classes at once using Generic Type Handler. For this, first of all, you have to implement `IGenericTypeProvider` interface. Write the code for all the required combinations for all data types that can be used in your application.

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to the Compact Serialization tab.
- Click Generic Type Handler button.
In the **Select Compact Classes** dialog, click **Browse** to select the assembly that contains your generic type provider class, for which you have implemented the `IGenericTypeProvider` interface.

All of the classes that had the implementation of the `IGenericTypeProvider` interface will get listed in **Class Name** drop down menu.

Select the required generic type provider class from the drop down list.

![Select Compact Classes](image)

Click **OK**. It loads all of the generic classes with their generic types.

You can remove any of the registered generic class at any time using the **Remove** button available at **Compact Serialization** tab.

Right click on the cache name in **Cache Explorer** and select the **Apply Configuration** option.
2.10. Registering Classes for Portable Data Sharing

There are situations when applications form the same/different platforms are interested to share cached data among them. For this purpose, dynamic data sharing can be used. It shares the data between various versions of an object, either from the same platforms or different platforms at runtime.

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to the **Data Sharing** tab.
- Click on **Add Shared Types** button.

```
Shared Types
```

- Select Data Sharing Classes wizard will open.
- Specify the name of data sharing type.
Click **Browse** to browse for required assembly (.exe, .dll or .jar file).
- Selected Assembly and all its classes will appear in **Loaded Classes** section.
- Select the required class and add it in **List of selected Classes** section using **Add Class** button.
- Repeat step 5 and 6 to select jar file and java class.
Click **OK** after adding classes of .NET and Java.

The opened wizard will close and selected classes of both platforms will appear in **Shared Types** section under your given data type name.
Now map the attributes of both platforms classes. Click on the added data share type ‘Customer’, all of its attributes will be listed in Common Attributes section.
• Click on a class listed under ‘Customer’ data type. Its attributes will be listed down in **Selected Class Attributes** section.

For portable types, please specify the attribute mapping from common attributes of an object to the version specific attributes. If attribute has no mapping available then leave it blank.

<table>
<thead>
<tr>
<th>Common Attributes</th>
<th>Selected Class Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>_CustomerID (System.String)</td>
<td>_CustomerID (System.String)</td>
</tr>
<tr>
<td>_CompanyName (System.String)</td>
<td>_CompanyName (System.String)</td>
</tr>
<tr>
<td>_ContactName (System.String)</td>
<td>_ContactName (System.String)</td>
</tr>
<tr>
<td>_Country (System.String)</td>
<td>_Country (System.String)</td>
</tr>
<tr>
<td>CustomerID (java.lang.String)</td>
<td></td>
</tr>
<tr>
<td>CompanyName (java.lang.String)</td>
<td></td>
</tr>
<tr>
<td>ContactName (java.lang.String)</td>
<td></td>
</tr>
<tr>
<td>Country (java.lang.String)</td>
<td></td>
</tr>
</tbody>
</table>

• A drop down will appear in the **Selected Class Attributes** section in front of each attribute of common attributes list. If .NET class is selected, select the required Java attribute of the Java class to map it against .NET attribute. Data types of attributes to be mapped should be same or compatible. The compatible .NET and Java data types mapping tables are given below:

<table>
<thead>
<tr>
<th>Primitive Types</th>
<th>Signed Types</th>
<th>Complex Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>sbyte</td>
<td>byte</td>
<td>DateTime</td>
</tr>
<tr>
<td>short</td>
<td>short</td>
<td>Date</td>
</tr>
<tr>
<td>int</td>
<td>int</td>
<td>ArrayList</td>
</tr>
<tr>
<td>int16</td>
<td>int</td>
<td>ArrayList</td>
</tr>
<tr>
<td>int32</td>
<td>int</td>
<td>HashMap</td>
</tr>
<tr>
<td>int64</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>float</td>
<td></td>
</tr>
<tr>
<td>double</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>decimal</td>
<td>BigDecimal</td>
<td></td>
</tr>
<tr>
<td>bool</td>
<td>boolean</td>
<td></td>
</tr>
<tr>
<td>char</td>
<td>char</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

**.NET class should be mapped with Java class, not vice versa.**

**.NET Class (Left side) ↔ Java Class (Right side).**
2.10.1. Import/Export Data Sharing Configuration

Using NCache Manager

Export Data Share Configuration:

- Click on the cache name in Cache Explorer to open cache settings.
- Go to the Data Sharing tab.
Click on the Export button and save the data sharing configuration in an xml file.

Provide a file name and click Save.

**Import Data Share Configuration:**

Click on the cache name in **Cache Explorer** to open cache settings.

Go to the **Data Sharing** tab.

Click **Import** to get data sharing configuration from an xml file.

Browse the saved data share configuration file and click **Open**.

Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.
You should have exported xml file containing mapping of attributes.

**Using Command Line Tool**

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
- Use `adddatashare` tool, provide required information:
  
  `adddatashare demoLocalcache -T C:/temp/SharedTypes.xml`

- Press ENTER, it will configure shared types from the provided xml file.
2.11. Configuring Data Source Providers

This section provides information about configuration of data source providers.

2.11.1. Configuring Read-Through Provider

Read-Through provider can be configured in following ways:

- **Using NCache Manager**
  - Click on the cache name in **Cache Explorer** to open the cache settings.
  - Go to Backing Source tab.
  - Check **Enable Read-Through** check box to enable it. After enabling it, other options will get enabled as well.
  - Click **Add** to select assembly that has **IReadThruProvider** implementation.
  - A new dialog box **Add New Provider** will open.
  - Enter **Provider Name** and click on ‘...’ button to browse for assembly implementing **IReadThruProvider** interface.
- Browse for the required assembly and select **Open**.
- Name of selected assembly will be appeared in **Assembly Name** and its classes implementing interface will be listed down in **Class Name** list box.
- Select the required Read-Through provider. You can also provide the values to your read-through provider class, if required. Just write the parameter name in **Parameter** field and its value against it under the **Value** field.

- Click **OK**.
These parameters along with their assigned values are passed to the selected Read-Through provider when it is initialized on cache start. Providing the parameters is optional.

- Selected provider class will be listed in Read-Through provider list at **Backing Source** tab.
You can configure multiple Read-Through providers for a cache.

- Now deploy the read-through provider assembly along with its dependency assemblies, if any. Click on the **Deploy Provider** button at the left bottom of the **Backing Source** tab.
- **Browse** for required assemblies.
- Select all the required assemblies and click **Open**.
- Apply the configurations by right clicking on cache name in **Cache Explorer** and selecting **Apply Configuration** option.

**Using Command Line Tool**

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
- Use `addbackingsource.exe`, provide the required information as:

```
addbackingsource.exe demoLocalCache /path:C:\Users\Administrator\Downloads\Providers\Providers.dll/cNCache.Sample.Providers.Readers.Reader1 /pnreader1/R
```
- Press **ENTER**, it will configure Reader1 read-through provider on demoLocalCache.
- Now deploy all of the required assemblies (assembly in which you have implemented the read-through provider and all of its dependency assemblies) using `deployprovider.exe`. Use following command on command prompt:
deployprovider demoLocalCache /T C:\Users\Administrator\Downloads\Providers\
  
  ● Press ENTER, it will deploy all of the assemblies from the given path to NCache deploy folder.

All deployed assemblies for a given cache reside on the following path:
%NCHOME%/deploy/your-cache-name/

2.11.2. Configuring Write-Through Provider

To preserve the cache data in some persistent storage, you can configure write-through providers for a cache. By using write through provider, the cached data and data added/updated/removed is synchronized with a data source.

Using NCache Manager

● Click on the cache name in Cache Explorer to open its settings.

● Go to the Backing Source tab.

● Check Enable Write-Through on this cache check box to enable it.

● Now click Add to select assembly implementing IWriteThruProvider interface.

● A new dialog box for Add New Provider will open.

● Enter Provider Name and click ‘...’ button to browse for assembly implementing IWriteThruProvider interface.
Browse for the required assembly and click **Open**.

Name of selected assembly will appear in **Assembly Name** and all of its classes implementing interface will be listed down in **Class Name** drop down menu.
Select the required Write-Through provider from the **Class Name** drop down list. You can also provide the values to your write-through provider class, if required. Just write the parameter name in **Parameter** field and its value in **Value** field.

- Click **OK**.
These parameters, along with their assigned values, are passed to the selected Write-Through provider when it is initialized on cache start. Providing the parameters is optional.

- Your selected write-through provider, along with the given parameters, is listed on **Backing Source** tab under **Write-Through** list box.
• Deploy the write-through provider assembly along with its dependent assemblies if any. Click on the **Deploy Provider** button at the bottom left of the **Backing Source** tab page. Browse for the required assemblies, select all of the required assemblies and click **Open**.

Deploy provider allows to copy/update provider assemblies to the cache nodes. On adding any read-through or write-through provider, use the ‘Deploy Provider’ button to select the appropriate assemblies for deployment.

• Apply the configurations by right clicking on cache name in **Cache Explorer** and selecting the **Apply Configuration** option.

**Using Command Line Tool**

• Open command prompt.

• Go to the NCache tools directory like: `cd %NCHOME%/bin/tools`

• Use `addbackingsource.exe`, provide following information:

```
addbackingsource.exe demoLocalCache /path C:\Downloads\Providers.dll/cNCache.Sample.Providers.Writers.Writer1 /pnwriter1/W
```

• Press **ENTER**, it will configure Writer1 as write-through provider on demoLocalCache.

• Deploy required assemblies using `deployprovider.exe`. Use following command on command prompt:

```
deployprovider demoLocalCache /T C:\Downloads\Providers.dll
```

• Press **ENTER**, it will deploy all assemblies from the given path to NCache.

### 2.11.3. Configure Write-ThroughProviders for Write-Behind Operations

If write-behind option is enabled, then all of the data source operations are queued up and performed in the background. There are multiple settings, which can be used to optimize the write-through provider to fulfill the user requirements.

**Using NCache Manager**

• Click on the cache name in **Cache Explorer** in **NCache manager**. Cache settings open in right side pane in NCache Manager.

• Go to the **Backing Source** tab.

• Configure a write-through provider as explained in ‘Configuring Write through provider’
Once here, you can change or specify the values as per your application requirement(s).

- Check the **Enable Batching** checkbox to perform all write-behind operations on data source in bulk/batches. You can also configure the **Batch-interval(s)** and **Operation-delay(ms)** for the batching mode.
- Change the value for **Operations per second**, to control how many write-behind operations will be performed on data source in a second.
- You can also configure the **Failed operations queue limit**. This queue is used to re-queue the failed operations and will be retried on data source.

- Right click on the cache name in **Cache Explorer** and select the **Apply Configuration** option.

### Using Command Line Tool

- Open command prompt.
- Go to the NCache tools directory like: cd %NCHOME%/bin/tools
- Use `addbackingsource.exe`, provide following information:

  addbackingsource.exe demoLocalCache /path
  C:\Users\Administrator\Downloads\Providers.dll /c
  NCACHE.Sample.Providers.Writers.Writer1 /pn writer1 /W /b true /ops 500

- Press **ENTER**, it will configure Writer1 as write-through provider on demoLocalCache with write-behind enabled and with throttling limit of 500 operations per second.
- Deploy required assemblies using `deployprovider.exe`. Use the following command on command prompt:

  Deployprovider demoCache/T C:\Downloads\Providers.dll

- Press **ENTER**, it will deploy all assemblies from the given path to NCache.
2.12. Configuring Cache Start-Up Loader

Cache start-up loader can be configured in following ways:

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open the cache settings.
- Go to the **Cache Loader** tab

Check the **Enable Cache Startup Loader** check box to enable it. Once it is enabled, other options will be enabled also.
Click on the ‘…’ button to select the assembly implementing ICacheLoader interface.

Browse for the required assembly and click **Open** after selecting the required assembly.

Name of selected assembly will appear in **Assembly Name** and its classes implementing interface will be listed down in **Class Name** list box in **Cache Loader** tab.

Select the required class.
You can also add the parameters for your cache loader implementation, if required.

- Click **New**.
- Provide the Parameter Name and Parameter Value in the New Parameter dialog box.

- Deploy the required assemblies using **Deploy Cache Loader** button at the lower left corner of **Cache Loader** tab.
- Browse for the required assemblies and select **Open**.

This step is optional and can be skipped. It is only required if your implementation of ICacheLoader requires parameter(s) on its initialization; otherwise there is no need to provide parameters for your cache loader.
Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.

### Using Command Line Tool

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`.
- Use `addstaruploader.exe`, provide the required information:

  ```
  addstartuploader demoLocalCache /a C:\Downloads\Providers.dll/cNCache.Sample.Providers.Loaders.Loader1
  ```
- Press **ENTER**, it will add Loader1 as startup loader on demoLocalCache.
- Deploy required assemblies using `deployprovider.exe`. Use following command on command prompt:

  ```
  deployprovider demoLocalCache /T C:\Downloads\Providers.dll
  ```
- Press **ENTER**, it will deploy all assemblies from the given path to NCache.
2.13. Configuring Cache Level Events

Cache Level Events can be enabled in two ways.

Using NCache Manager

- Click on the cache name in Cache Explorer to open cache settings.
- Go to the Options tab.
- Check Item add notification, Item remove notification and Item update notification check boxes.

<table>
<thead>
<tr>
<th>Notifications</th>
<th>Cache event notifications let you design applications that react to various cache events. Event notifications are disabled by default to save precious cpu and network resources. Select the notifications that your application uses.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item add notification. (OnItemAdded)</td>
<td></td>
</tr>
<tr>
<td>Item remove notification. (OnItemRemoved)</td>
<td></td>
</tr>
<tr>
<td>Item update notification. (OnItemUpdated)</td>
<td></td>
</tr>
</tbody>
</table>

- Right click on the cache in Cache Explorer and select Apply Configuration option.

2.14. Configuring Eviction Policy

If a cache reaches its maximum storage limit and eviction is turned on, the specified ratio of items will be evicted to make space for new objects. On the other hand, if cache reaches its maximum storage limit and eviction is turned off, items are neither evicted nor added to the cache any further.

An eviction scheme is used to select the objects that need to be removed from cache. A particular eviction policy can be selected at the creation time but once it is set it cannot be altered in the running cache. User can change eviction policy after creating a cache only if it is not running. Three types of Eviction Policies are provided by NCache:

1. Least Frequently Used

In this scheme a counter that counts the access made to an object is used. At the time of eviction objects with lowest counter number are evicted first. Here too a random selection is made between two objects with same counter.

2. Least Recently Used

This eviction scheme associates a timestamp with all objects. This timestamp is updated each time the object is accessed. When evicting, objects with the oldest timestamp are picked first. In case of two objects with the same time stamp, a random selection is made.

3. Priority Based Eviction

This is the default eviction policy and has the same mechanism as LFU with the addition of a relative cost that is associated with an object at the time of its addition to cache. Objects with lower cost are removed before objects with higher cost. Client application can choose from a list of priorities that is predefined (Low, Below Normal, Normal, Above Normal, High and Not Removable). In case a priority is not chosen, it is set by default as Normal.

These eviction policies for a cache can be specified in one of the following two ways.
Using NCache Manager

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to the **Policies** tab.
- Uncheck the **Turn off Eviction** check box.

![Eviction Policy](image)

- **Eviction Policy**
  Customize the way cache will remove existing items to accommodate new items. Existing items will be removed based upon any of the following eviction strategies.

  - **Eviction Policy:**
    - Least Recently Used
  - **Default Priority:**
    - high
  - **Eviction Percentage:**
    - 5

  - **Turn off Eviction.**

- Select the **Eviction Policy** from drop down list.

  ![Eviction Policy](image)

  - **Eviction Policy:**
    - Least Recently Used
  - **Default Priority:**
    - Least Frequently Used
  - **Eviction Percentage:**
    - Least Recently Used

  - **Turn off Eviction.**

- If you have selected the **Priority based** eviction, then **Default Priority** can also be changed in the drop down list.

  ![Eviction Policy](image)

  - **Eviction Policy:**
    - Priority Based
  - **Default Priority:**
    - high
  - **Eviction Percentage:**
    - High
      - above-normal
      - normal
      - below-normal
      - low

- Change the **Eviction Percentage** if required.
2.15. Configuring Compression

By default, cache is created with compression disabled. However, compression can be enabled as follows:

- **Using NCache Manager**
  - Click the cache name in **Cache Explorer** to open cache settings.
  - Go to the **Options** tab.
  - Check the **Enable Compression** checkbox.
  - Change the **Threshold size** as per your requirement.
  - Right click on the cache name in **Cache Explorer** and select the **Apply Configuration** option.

2.16. Configuring Email Notifications/Alerts

You can configure email notifications in NCache. NCache will send you an email automatically when a selected event occurs in cache for which you have enabled notification. You can configure email notification for a cache in the following way:

- **Using NCache Manager**
  - Click on the cache name in **Cache Explorer** to open cache settings.
  - Go to the **Alerts** tab.
  - Check the **Enable Email Notifications** checkbox, to enable email notification feature.
  - Enter sender email in the **Sender Email** text box.
  - Enter SMTP server in the **SMTP Server** text box.
  - Enter Port in the **Port** text box.
  - Check **SSL Required** check box, if your email server requires SSL authentication.
- Check the **Authentication Required** check box if required and enter login and password in **Login** text box and **Password** text box respectively.
- Specify recipients’ email addresses in **To** text box.
- Check the cache events for which you want to receive notifications from NCache.

- Apply configuration by right clicking on the cache name in **Cache Explorer** and selecting **Apply Configuration** option.
2.17. Binding Cluster and Client/Server with Dedicated IP for Performance

NCache provides you with the opportunity to bind cluster, client/server or both (cluster and client/server) with dedicated IP addresses.

2.17.1. Binding Cluster with a Dedicated IP

Cluster can be bound with a dedicated IP address in the following way:

**Using NCache Manager**

- Right click on a server IP address under **Server Nodes** in Cache Explorer and click on **Select NIC** option.

- **Available Network Interface Cards** dialog will open. It will contain all network cards available on that node with drop down list.
• By default IP is used for both client-server and inter cluster communications.

• Select the Cluster option from the drop down list in front of the IP address with which you want to bind the cluster.

• Click Apply.

• NCache will ask for confirmation to apply configuration and restart service. Click Yes.
This restarts the NCache service to use the selected IP addresses for the specified communication (inter-cluster).

2.17.2. Binding Client/Server with a Dedicated IP

Client/server can be bound with a dedicated IP address in the following way:

Using NCache Manager

- Right click on a server IP address under Server Nodes in Cache Explorer and click on Select NIC option.

Available Network Interface Cards dialog will open. It will list all network cards available on that node with their assigned IP addresses.

- Select the Client/Server option from drop down in front of the IP with which you want to bind the client and server.
Click Apply.

NCache will ask for confirmation to apply configuration and restart service. Click Yes.

This restarts the NCache service to use the selected IP addresses for the specified communication (client-server).
2.18. Configuring MapReduce

You can configure MapReduce for processing and generating large data sets with a parallel, distributed algorithm on a cluster.

Using NCache Manager

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to the **MapReduce** tab.
- Click **Deploy Task Libraries**.

![Deploy Task Libraries dialog box](image)

- A dialog box will open. Browse for the libraries which have the MapReduce interfaces implemented and click **Open**.
- A Notification will appear after successful deployment of assemblies.

![Notification](image)

- Maximum number of MapReduce tasks to be executed simultaneously can be changed according to your requirements.

![Max Tasks](image)

- In case you expect exceptions to be thrown during task execution, you can specify the number of exceptions to be avoided from your code, after which the task is failed and logged in the cache error log.

![Max. avoidable exceptions](image)

- You can modify chunk size - the number of tasks processed in the Mapper and Combiner - before transmitting to Combiner or Reducer.
You can modify the maximum number of tasks that can wait in queue before they are processed.

2.19. Configuring Heartbeat

Communication between Cluster nodes is critical for smooth cluster operations. Therefore, NCache provides an option called Heart Beat for the clusters to detect connection breakage with any node of the cluster. Connection breakage may occur due to an unplugged network cable or disabling of a network connection. When this option is enabled, a node of a cluster sends some dummy packets to other nodes and waits for acknowledgment. Therefore, if live nodes do not receive acknowledgments from any idle node after specific duration, they will consider it dead and will update their views by excluding it from the cluster. There will be no overhead of heart beat messages when all nodes are active and sending messages to each other.

Using NCache Manager

You can enable or disable heartbeat option from **Cluster Settings** Tab of Cluster view window.
2.20. Add Existing Clustered Caches

**Using NCache Manager**

- Right click on the **Clustered Caches** in Cache Explorer.
- Click on the Add Existing Clustered Cache option.

- Specify the node IP address in text box or select from the **Network Neighborhood** listed under **Available Nodes** list box and click on the **Next** button.

- List of all the clustered caches registered on the specified node are displayed.
• Select the cluster and click **Finish**.

• The selected cache will appear under **Clustered Caches** tree node in **Cache Explorer**.
3. **Management Operations**

   This section describes how you can manage caches and their related configuration.

3.1. **Start Cache**

   You can start a clustered cache on all the server nodes altogether or start it on a specific node. Please follow these steps to start the clustered cache:

3.1.1. **Starting Cache on All Server Nodes**

   **Using NCache Manager**

   - From the **Cache Explorer** view, right click on the cache name, which you want to start.
   - Click on the **Start** option in the right click context menu.

   ![Cache Explorer](image)

   - This starts the demoClusteredCache on all the server nodes listed under the **Server Nodes** tree view. Server node icon color turns to sky blue if cache is started successfully.
3.1.2. Starting Cache on a Particular Server Node

Using NCache Manager

- From the Cache Explorer view, right click on the server name under Server Nodes tree node, on which you want to start the cache.
- Click on the Start option in the right click menu, in order to start the cache.

This starts the demoClusteredCache on the selected server node. Server node icons turn to sky blue if cache is started on it.
Using Command Line Tool

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
- Use `startCache.exe`, provide required information:
  ```
  startcache.exe demoClusteredCache /s 20.200.21.95 /p 8250
  ```
- Press ENTER, it starts the demoClusteredCache on server node 20.200.21.95.
3.2. **Stop Cache**

You can stop a clustered cache on all the nodes altogether or stop it on a specific node.

3.2.1. **Stop Cache on all Server Nodes**

**Using NCACHE Manager**

- From the **Cache Explorer** view, right click the clustered cache on which you want to stop the cache.
- Click on the **Stop** option.

- This will stop the demoClusteredCache on all the server nodes added in this cache. If cache is stopped successfully, then the color all of the server nodes change to black, like this:
3.2.2. Stopping Cache on a Particular Server Node

Using NCache Manager

- From the Cache Explorer view, right click on the server name under Server Nodes tree node, on which you want to start the cache.
- Click on the Stop option in the right click menu.

This will stop the demoClusteredCache on the selected node. Icon color of the selected node changes to black if cache is stopped successfully on it, like this:
You can stop a node gracefully using **Graceful Stop** option. The graceful stop option makes sure that all of the queued operations are performed before cache is stopped.

**Using Command Line Tool**

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
- Use `stopcache.exe`, provide required information:
  ```
  stopcache.exe demoCache /s 20.200.21.95 /p 8250
  ```
- Press **ENTER**. It stops the cache named demoClusteredCache on the specified server 20.200.21.95.
3.3. Manage Cache Service on a Server Node

You can start/stop/restart NCache service on a server node added to a clustered cache by following the steps given below:

3.3.1. Start Cache Service on a Server Node

Using NCache Manager

- In Cache Explorer view, right click the server node on which you want to
- Click on the Start Cache Service option in the right click context menu.

This will start the NCache service on this node.

3.3.2. Stop Cache Service on a Server Node

Using NCache Manager

- In Cache Explorer view, right click on the server node on which you want to stop the service.
- Click on the Stop Cache Service option in the right click context menu.
This will stop the NCache service on this selected node.

3.3.3. **Restart Cache Service on a Server Node**

**Using NCache Manager**

- In **Cache Explorer** view, right click on the server node on which you want to restart the cache service.
- Click on the **Restart Cache Service** option in the right click context menu.
This will first stop the NCache on this node and then restart it.
3.4. Apply Configuration

You change the settings of a cache while it is stopped; follow these steps in order to apply the configuration changes:

- From the Cache Explorer view, right click on the cache name to which you want to apply configuration.
- Make sure the cache is stopped, else the option will be greyed out.
- Click on the Apply Configuration option.

- Your changes will be applied on all of the nodes of this clustered cache.

Through Toolbar Icon

You can also apply configurations using the option on the toolbar.
3.5. **Hot Apply Configuration**

Configuration changes made to a cluster while it is running can be applied using **Hot Apply Configuration**. For this, please follow the steps given below:

- From the **Cache Explorer** view, right click on the cache name to which you want to apply configuration.
- Make sure the cache is stopped, else the option will be greyed out.
- Click on the **Hot Apply Configuration** option.

Your changes will be applied to all the nodes of this clustered cache.

There are only a few settings that you can change while the cache is running, e.g., you can increase or decrease the cache size of a running cache.

**Through Toolbar Icon**

You can also hot apply configurations using the option on the toolbar.
3.6. Reset Configuration

Resetting configuration will change all configuration changes to the last saved configuration values for the current cache.

- From the Cache Explorer view, right click the clustered cache whose configurations you want to reset.
- Make sure the cache is stopped, else the option will be greyed out.
- Click on the Reset Configuration option.

- Cache settings are reset to the last saved configuration for this cache.

Through Toolbar Icon

- You can also hot apply configurations using the option on the toolbar.
3.7. Data Load Balancing

You can request to balance data on a node in a partitioned cluster without waiting for the automatic load-balancing task to trigger it. This brings data load on the node near to average data per node while the rest of the data is distributed across other nodes in the cluster. Candidate nodes for accepting the load are selected on the basis of the amount of data currently present on them. A node having less data gets a larger share from the load.

- From the Cache Explorer view, right click a server node of a cache cluster.
- Click on the Balance Data Load option.
- Data will be evenly distributed across the nodes of the cluster.

Auto Load Balancing

- Go to the Auto Load Balancing tab in NCache Manager.
- Check the Enable Auto Load Balancing option and configure Threshold and Interval according to your requirements.
Situations can arise when some of the nodes in the cluster are over-populated whereas the storage capacity of rest of the nodes is not fully utilized. An automatic data load balancing task can be enabled that will periodically look for such situations and will try to balance the data load among other nodes.

**Enable Auto Load Balancing.**

Threshold is the percentage data size above the average data size per node. A threshold of 16% means that the data load will be balanced when a node gets 16% more data than the average data size per node.

**Balancing Threshold:** 50

Auto balancing task will look for the need of data load balancing periodically after this interval.

**Balancing Interval (sec.):** 50
4. Configuring Cache Clients

This section describes step-by-step instructions on how to add or remove client nodes in a clustered cache, create or remove client cache and configure client specific settings. A client node is the one from where clustered cache can be initialized to make API calls. You can add client node in the clustered cache in either of the two ways described below:

4.1. Adding Client Node

Using NCache Manager

Each clustered cache tree in Cache Explorer contains a node known as Client Nodes. Any number of client nodes can be added in a clustered cache.

- Right click on Client Nodes in Cache Explorer for a cache and then click on the Add Node option.
- Provide the client machine name/IP by expending Network Neighbourhood or you can manually enter client node name/IP in text box.
If you have more than one IP address on a client machine, then enter the IP address on which NCache service is running.

- Click on the **Finish** button, newly added client node appears under **Client Nodes** in **Cache Explorer** tree view.

```
Cache Explorer

- Cache Management
  - Clustered Caches
    - democlustersedcache
  - Server Nodes
    - Client Nodes
      - 20.200.21.144
  - Client Cache
    - cache2
  - Local Caches
    - 20.200.21.95
      - demolocalcache
  - Bridges
```

Having problems browsing network?
Change Security Options
Using Command Line Tool

You can use addclientnode tool to add client node in a cache. This tool is shipped with NCache.

- Open command prompt.
- Go to the NCACHE tools directory like: cd \%NCHOME\%/bin/tools
- Using addclientnode.exe provides the required information like:

  addclientnode.exe demoClusteredCache /s 20.200.21.95 /e 20.200.21.144

- Press the ENTER key, it adds 20.200.21.144 as client node in cache named demoClusteredCache.

4.2. Configuring Client Node Settings

There are some configuration settings related to client node. Those can be configured using NCACHE Manager.

- Click on a client machine IP address in **Cache Explorer** for a cache to open client related settings of this client node.

- On Main tab you can configure Connection Timeout, Connection Retries, Retry Interval and Client Request Timeout.
- All values are in seconds. A brief description of each attribute is given is given in NCACHE Manager alongside each attribute.
Using the **Settings** tab, you can:

I. Enable/disable client load balancing using the checkbox **Enable Client Load Balancing**, 
II. Specify **Server Priority** for this client,
III. Change **Sync Mode** of client cache to either **Optimistic** or **Pessimistic**.
Using the Security tab you can specify Primary and Secondary user credentials, i.e., User Id and Password. These credentials will be used when this cache is initialized from this client node.

At the Backing Source tab, you can select the default Read-Thru and/or Write-Thru providers, only if these providers are configured for the cache.
- Apply configuration changes by right clicking on the cache name in Cache Explorer and select Apply Configuration option.

4.3. Creating Client Cache

Client Cache is a cache that exists on client node. You can create client cache in either of the two ways described below:

Using NCache Manager

- Right Click on the Client Cache tree node in Cache Explorer for a cache. Click on the Create New Cache option in the right click menu.
  
  - Enter client cache Name and select client cache Sync Mode.
  - Click Next.
Select the required **Isolation Level**, it changes the **Data Format** accordingly.
• In **Advanced options** dialog, you can configure following:
  
  o You can change the **Eviction policy** and **Eviction percentage**; items will be evicted from cache using the selected policy and percentage value, if eviction is enabled.
  
  o You can change **Clean interval** value. Its default value is 15 sec.
  
  o You can make the cache to start automatically after service restart by checking checkbox **Auto start this cache on service startup**.

![New Client Cache Wizard](image)

- This will create client cache on all client nodes and **Client Cache Creation Status** window is displayed showing client cache creation status for each client node.
Click Close and the newly created client cache will appear under Client Cache in the Cache Explorer.

Using Command Line Tool

You can use createclientcache tool to create client cache. This tool is shipped with NCache.

- Open command prompt
- Go to the NCache tools directory like: cd %NCHOME%/bin/tools
- Using createclientcache.exe provide the required information like

- Press the ENTER key, it will create a client cache demoClientCache for cache demoClusteredCache on node 20.200.21.144.

4.4. Disabling Client Cache on Client Nodes

In some cases when you create client cache but do not want to use it, then you can disable client cache for a particular client node or for all client nodes.

Each client node has an indicator showing whether client cache is enabled or disabled. A red square sign indicates that client cache is enabled and grey square sign indicates that client cache is disabled, as shown in screen shots below.

Using NCache Manager

- Right click on Client Nodes, context menu appears.

- Now click on Disable Client Cache option. Client cache becomes disabled for all the client nodes, currently listed under Client nodes in Cache Explorer.

- Grey square sign on client node indicates a disabled client cache.
Please note that you can also disable client cache by right clicking on a particular client node instead of **Client Nodes**. In this way, client cache will be enabled for the selected client node only.

### 4.5. Enable Client Cache on Client Nodes

If a client cache is already created but currently is disabled, you can enable this client cache at any time.

**Using NCache Manager**

- Right click on the **Client Nodes** in **Cache Explorer** for a cache and click on the **Enable Client Cache** option.

- Client cache becomes enabled for this cache in **Cache Explorer**. Red square sign on client node indicates that client cache is enabled now.
Please note that you can also enable client cache by right clicking on particular client node instead of Client Nodes in Cache Explorer. In this way, client cache will be enabled for the selected client node only.

4.6. Configuring Client Cache Settings

This section explains how to configure some basic settings of the client cache.

4.6.1. Configuring Synchronization Mode

There are two synchronization modes for a client cache.

**Optimistic Synchronization**

In optimistic mode, application fetches data from client cache despite the fact that data is updated or not. It assumes that data is newer in client cache. In optimistic synchronization, updates are propagated to the client cache using event notifications.

**Pessimistic Synchronization**

Pessimistic mode is used when applications are handling sensitive data. On data fetch request, if data is found in client cache, its version is matched with item on remote cache. If both versions are same, then item is returned otherwise updated item is fetched from remote cache and is updated back to client cache.

These modes can be configured as follows:

**Using NCache Manager**

- Click on a client node IP under Client Nodes in Cache Explorer. Client specific settings in tabbed view shown in right pane.
- Go to **Settings** tab and select any one option **Optimistic** or **Pessimistic** from **Sync Mode** Drop Down list.
4.6.2. Configuring Client Cache Size

- Client cache size is configurable. Click on the client cache and go to the Main tab in the right pane. Under this tab, Cache Max. Size can be changed.
Right click on client cache and select **Apply Configuration** option.

### 4.6.3. Configuring Isolation Level (InProc/OutProc)

Client cache can be InProc or OutProc. These levels can be configured as follows:

**Using NCache Manager**

- Click on client cache under **Client Cache** tree view in **Cache Explorer** and go to the **Main** tab in the right pane.
- In **Main** tab, a drop down list labeled as **Isolation Level** is available. The levels can be changed to either **High(OutProc)** or **Low(InProc)**.
Right click on client cache name in **Cache Explorer** and select **Apply Configuration** option.
5. Configuring Security

Every Server node keeps security information that it has to provide to calling clients. Alachisoft.NCache.NCSecurity.dll at server node is required to provide security services to all clients. Whatever information is required by this assembly, for managing security related issues, is present in security.ncconf file. security.ncconf file helps enabling NCache security as well as to manage Node and API level security. You will find security.ncconf file at "%InstallDirectory%/NCache/config".

security.ncconf file is as shown below:

```xml
<cache-security>
  <enabled>false</enabled> <!-- Change the value true/false to enable or disable security feature in NCache -->
  <!-- LDAP path is required to log-on to domain. For more information see http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adsi/adsi/ldap_adspath.asp -->
  <ldap>LDAP://DomainControllerName</ldap>

  <!-- Following users are Administrators and are allowed to add 'this' node into their clusters and also allowed to perform management operations like 'Start Cache', 'Stop Cache', 'Remove Cache' etc-->
  <ncache-administrators>
    <!-- List of Administrators -->
  </ncache-administrators>
</cache-security>
```
5.1. Configuring Security for Cache Management

After you have created cache, security for management operations can be configured in following ways:

**Using NCache Manager**

- Open NCache Manager.
- Go to **Security** menu option and select **Security Credentials** option.

Security Credentials dialog will open.

Settings on this tab will also help browse for nodes in the network.

**Domain Info**

- **Domain Controller:**
- **Port:** leave empty for default setting.

**Logon Credentials**

- **User Name:**
- **Password:**
• Provide the valid credentials for your active directory like name of Domain Controller, Port (optional), User Name and Password.

![Security Credentials](image)

If security is enabled, one has to provide credential information to perform operations on Clusters and Local Caches in NCache Manager. Information given here will be cross checked with security.nconf file each time user verification is required for managerial operations.

Settings on this tab will also help browse for nodes in the network.

<table>
<thead>
<tr>
<th>Domain Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Controller:</td>
</tr>
<tr>
<td>Port:</td>
</tr>
<tr>
<td>- Leave empty for default setting.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logon Credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name:</td>
</tr>
<tr>
<td>Password:</td>
</tr>
</tbody>
</table>

![Verify Credentials](image)

• Click on the Verify Credentials button. It will prompt success or failure messages. If provided security credentials are correct then it will display the success message like this:

![User credentials verified](image)

• Click OK.

• Now go to NCache Manager -> Security menu option again and select Configure Security option.
- **Security Configuration Wizard** will open. Specify the server IP address for which you want to configure the security and click **Next**.

- Security Configurations Wizard will open.

- Expand the **Available Domain Users**. It will list down all of the active directory users.
Select users from the users list and then add the required users using '>' button. These users will be allowed to manage the caches.
- Check the **Enable Security** checkbox to enable the security.

Using Command Line Tool

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
- Use `configurenodesecurity.exe`, provide the required information like:

  ```
  configurenodesecurity.exe /E /n 20.200.21.95 /administrator /P yourpassword
  ```

- Press **ENTER**, it will add administrator in `security.ncconf` on 20.200.21.95 node and it will be allowed to add management operations.

- If you want to give access to more users, use following command after step 2:

  ```
  configurenodesecurity.exe /A /N james_tredwell /n 20.200.21.95 /U administrator /P yourpassword /C yourdomain.org
  ```

- Press **ENTER**, it will add james_tredwell in secured users list for management operations.
5.2. Configuring Security for Cache

Security for cache can be configured on following ways:

Using NCache Manager

- Open NCache Manager.
- Go to Security menu option and select Security Credentials option.
- Security Credentials dialog will open.

If security is enabled, one has to provide credential information to perform operations on Clusters and Local Caches in NCache Manager. Information given here will be cross checked with security.nccomfile each time user verification is required for managerial operations.

Settings on this tab will also help browse for nodes in the network.

Domain Info

- Domain Controller: 
- Port: - Leave empty for default setting.

Logon Credentials

- User Name:
- Password:
• Provide the valid credentials for your active directory like name of **Domain Controller**, **Port** (optional), **User Name** and **Password**.

![Security Credentials](image)

If security is enabled, one has to provide credential information to perform operations on Clusters and Local Caches in NCache Manager. Information given here will be cross checked with security.nconfig file each time user verification is required for managerial operations.

Settings on this tab will also help browse for nodes in the network.

<table>
<thead>
<tr>
<th>Domain Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Controller:</td>
</tr>
<tr>
<td>Port:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logon Credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name:</td>
</tr>
<tr>
<td>Password:</td>
</tr>
</tbody>
</table>

• Verify the credentials using **Verify Credentials** button. It will prompt success or failure message in case of success or failed verification of credentials successfully.

![NCache Manager](image)

• Click **OK** if credentials are successfully verified.

• Click the cache name for which you want to configure the security. Cache settings open in the right pane inside the NCache Manager.
- Go to the **Security** tab in cache settings.

- Expand the domain users under the **Available Domain Users** section.

  Cache level security allows you to specify cache users, who can make API calls like InitializeCache, Add, Get etc.

  - **Available Domain Users:**
    - Domain Users
      - test.org
      - aaron_finch
      - Administrator
      - alex_hales
      - chris_jordan
      - glenn_mawwell
      - Guest
      - james_tredwell
    - **Selected Users:**

  At least one User must be provided to Enable Security

  - **Enable Security**

- Add required users to **Selected Users** section using ‘>’ button. These users will have access for cache operations.

  Cache level security allows you to specify cache users, who can make API calls like InitializeCache, Add, Get etc.

  - **Available Domain Users:**
    - Domain Users
      - test.org
      - aaron_finch
      - Administrator
      - alex_hales
      - chris_jordan
      - glenn_mawwell
      - Guest
      - james_tredwell
    - **Selected Users:**

  At least one User must be provided to Enable Security

  - **Enable Security**

- Check **Enable Security** check box.
Cache level security allows you to specify cache users, who can make API calls like InitializeCache, Add, Get etc.

Available Domain Users:
- Domain Users
  - test.org
    - aaron_finch
    - Administrator
    - alex_hales
    - chris_jordan
    - glenn_maxwell
    - Guest
    - james_tredwell

Selected Users:
- aaron_finch
- alex_hales

At least one User must be provided to Enable Security

You can also disable management security by unchecking the **Enable Security** check box at any time.

- Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.

### Using Command Line Tool

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`
- Using `configurecachesecurity.exe`, provide the following information:
  ```
  configurecachesecurity demoClusteredCache /E /s20.200.21.95 /U james_tredwell /P yourpassword /C yourdomain.org
  ```
- Press **ENTER**, it will enable security for demoClusteredCache on server 20.200.21.95 and add the user james_tredwell.
- If you want to add more users, use the following command after step 2:
  ```
  configurecachesecurity demoClusteredCache /A /N aaron_finch /s 20.200.21.95 /U administrator /P yourpassword /C yourdomain.org
  ```
- Press **ENTER**, It will add the user aaron_finch to the secured users list of demoClusteredCache.
5.3. Configure Encryption for a Cache

Encryption can be configured on cache in the following way:

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to the **Security** tab.
- Check the **Enable** checkbox inside **Encryption** group to enable encryption.
- Select the encryption provider as per your requirement from the given **Provider** dropdown list.
  
  ![Encryption Configuration](image1)

- Enter the encryption key in the **Key** textbox.

  ![Encryption Configuration](image2)

  Different encryption providers require different key lengths.

- Right click on the cache name in **Cache Explorer** and select **Apply Configuration** option.
6. Configuring Bridge for WAN Replication

This section describes step-by-step instructions on how to create and configure a bridge in NCache.

6.1. Overview of Bridge

Remote data replication is a critical component for any plan to ensure effective and efficient protection of data and rapid recovery from a major interruption. Synchronous replication of data is good internally for the cluster but its impact on performance becomes a significant consideration when two clusters of cache are geographically separated. Bridge topology is designed for the scenarios, which involve replication of data from one on-site cache(s) to another on-site/off-site cache(s) across the WAN for disaster recovery. Due to asynchronous replication, all clients connected to the active cache(s) get an impression that the operations are being performed on the active cache while a complete backup is taken to the other cache(s) seamlessly.

When an operation is performed on the source cache, it is asynchronously handed over to the bridge cache. This operation is then queued in a queue maintained by the bridge cache. Operations from the queue are transferred to the target cache when bridge finds the target cache available and ready to accept operations. With the bridge topology, it is ensured that:

- There is no performance degradation.
- Operations are performed in same sequence as they were on original cache.
- Operations are not lost in case of connection failure.

NCache provides the following Bridge topologies:

- Active-Passive (one Active and one Passive Cache)
- Active-Active (two Active Caches and no Passive Cache)

6.1.1. Active-Passive

This bridge topology has one active and one passive cache.

Three NCache Clusters participate in Active-Passive topology:

1. An **active cache** (replicated, partitioned, mirrored or partitioned-replica), where all clients connect and perform read and write operations.

2. A **passive cache** (replicated, partitioned, mirrored or partitioned-replica), where all operations performed on the active cache are replicated. Clients can connect to passive cache and perform both read and write operations but those operations are not replicated to the other cache. Modifications can be done at passive cache if required.

3. A **bridge cache** between the active cache and the passive cache.
6.1.2. Active-Active

This bridge topology has two active caches but no passive cache. Three NCache Clusters participate in the Active-Active topology:

1. Two active caches (replicated, partitioned, mirrored or partitioned-replica), where all clients connect and perform read and write operations. Here any updates to the cache are applied asynchronously to the other cache.

2. A bridge cache between both the active caches.
6.2. Creating Clustered Bridge

You can create a bridge for NCache using the following tools.

**Using NCache Manager**

- From the *Cache Explorer* view, right click the *Bridges* node and click *Create New Bridge* option.

- The New Bridge Wizard appears, displaying the *Specify Bridge Identification* dialog.

- Provide the **Name** for the new bridge and click **Next**.

  ![New Bridge Wizard](image)

- Please make sure that the bridge name you provided is unique on all nodes that are a part of this bridge.

- Select Bridge Member Nodes dialog opens up.
- Specify the node IP address in text box or select from the Network Neighborhood and click on the '>' button to add it in selected nodes list.

- Click Next once all required nodes are added.
You can add a maximum of two nodes as bridge members.

- **Specify Queue Parameters** dialog box appears. You can specify the multiple bridge properties, i.e., **Bridge Port**, **Maximum Queue Size**, and **Replicator Queue Size**.
After providing the required values click **Next**.

On the Specify Target Cache Options dialog box, you can change the Queue Replication Interval and Connection Retry Interval values.

There is also a check box **Start this bridge on Service restart**. Check it if you want to start this bridge automatically when NCache bridge service starts.

Click **Finish**.
Your bridge will be successfully created and will be visible in the Cache Explorer view under the Bridges node.
6.3. Adding Clustered Caches to the Bridge

Depending on your bridge topology, you can add two caches to your bridge. The caches that you want to add should already be created on the local server or on remote server(s). Please follow these steps to add a cache to your bridge:

Please stop a running cache first if you want to add this cache in the bridge.

- From the Cache Explorer view, right click the bridge to which you want to add a cache and then click on the Add Clustered Cache option.

- On the Add Clustered Cache wizard, specify the server IP address in text box or select from the Network Neighborhood listed in Available Nodes list box.

- Click Next.
- **Select a clustered cache** dialog window will appear. It contains a list of all of the caches on your provided server.
- Choose any one cache and click **Next**.
On the **Select settings for bridge clustered cache** dialog, choose the cache mode for this bridge configuration. Default cache mode is **Active**, but you can change that.

The selected cache will successfully be added for this bridge and will be visible under the **Caches** node of the bridge.

Add the second cache for this bridge using same steps mentioned above, here you can specify the topology (active-active/passive), based on the mode you choose.
6.4. Configuring Bridge Settings

The NCache Manager provides the ability to change the configuration of bridges, whether it’s **Active-Passive** or **Active-Active**. To change the configuration of a bridge, click the bridge from the **Cache Explorer** view. You will be able to see various tabs in the right hand pane through which you can configure various options of your bridge.

6.4.1. Configuring Bridge Queue Settings

Bridge stores the received data/objects in cache queues. You can modify (increase or decrease) these queues’ size.

**Using NCache Manager**

- Click on the bridge name in **Cache Explorer** to open bridge settings.
- Go to the **Main** tab.

<table>
<thead>
<tr>
<th>Bridge Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bridge Name:</strong> demoBridge</td>
</tr>
<tr>
<td>Each node in bridge listens on it’s specified Port to accept connections from other bridge nodes. Every Bridge must have a unique Port to allow the existence of multiple bridges created on same node.</td>
</tr>
<tr>
<td><strong>Bridge Port:</strong> 10001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bridge Replicator Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify each Replicator Queue Size (Virtual Size).</td>
</tr>
<tr>
<td><strong>Replicator Queue Size:</strong> 500 MB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bridge Queue Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify an upper limit for the size of the queue in MB. The amount of items in queue is limited by maximum queue size and amount of free memory (physical and virtual).</td>
</tr>
<tr>
<td><strong>Queue Size:</strong> 500 MB</td>
</tr>
</tbody>
</table>

- Enable Queue Optimization.

- Increase or decrease the **Queue Size** according to your requirements.

You can also update the **Replicator Queue Size** on this tab.

- Right click on the bridge name in **Cache Explorer** and select **Apply Configuration** option.
6.4.2. Configuring Bridge Connection Settings

Every bridge cache opens a unique port for communication with other bridge nodes.

**Using NCache Manager**

- Click on the bridge name in **Cache Explorer** to open bridge settings.
- Go to the **Main** tab.

![Bridge Configuration Screen](image)

- Change the **Bridge Port**. Make sure that your newly provided port is not already in use of any other application.
- Click on the **Bridge Caches** tab.
- Change the **Connection Retry Interval**, which suits your environment (according to WAN latency).
Right click on the bridge name in Cache Explorer and select **Apply Configuration** option.

### 6.4.3. Enable Queue Optimization

NCache provides the optimized queue to eliminate the redundant operation replication to the bridge and other caches.

**Using NCache Manager**
- Click on the bridge name in **Cache Explorer** to open bridge settings.
- Go to the **Main** tab.
6.4.4. Configuring Bulk Size for Operation Replications

Replication can be controlled or throttled according to the underlying network performance to boost the bridge replication performance.

**Using NCache Manager**

- Click on the bridge name in **Cache Explorer** to open bridge settings.
- Go to the **Bridge Caches** tab.

- Check the **Enable Queue Optimization** check box on this tab.
- Right click on the bridge name in **Cache Explorer** and select **Apply Configuration** option.
6.4.5. Performance Counters

NCache provides and publishes multiple performance counters for bridge caches. You can enable or disable the performance counters publishing according to your requirements.

**Using NCache Manager**

- Click on the bridge name in **Cache Explorer** to open bridge settings
- Go to the **Options** tab
- Check or uncheck the Enable performance counters check box.
- Right click on the bridge name in **Cache Explorer** and select **Apply Configuration** option.
6.5. Configuring Conflict Resolver

There are situations when a bridge tries to replicate an operation for a key that already exists in the cache. Conflict Resolver is introduced to resolve the conflict between existing operations and operation replicated from bridge to the cache. It decides which key to apply on the cache depending upon defined rules. By default the decision will be made on **Time Based** rule, which keeps the latest key on a cache. Users can also define their own rules for conflict resolver.

**Using NCache Manager**

- Click on the cache name in **Cache Explorer** to open cache settings.
- Go to the **Conflict Resolver** tab.
- Check **Enable Conflict Resolver** check box to enable it. Others options will become enable after that.

![Conflict Resolver Configuration](image)

Conflict Resolver is used to resolve the conflict between existing operations and operation replicated from bridge to cache. Conflict arises when we try to replicate an operation from bridge for a key that already exist in cache. It decides which operation to apply on the cache depending upon rules implemented in Conflict Resolver. By default it is time based Conflict Resolution which means we only apply operations from bridge that are latest as compared to existing operations in cache.

- Click on the ‘…’ button and select the assembly in which you have implemented the **IBridgeConflictResolver** interface.
**Conflict Resolver** is used to resolve the conflict between existing operations and operation replicated from bridge to cache. Conflict arises when we try to replicate an operation from bridge for a key that already exist in cache. It decides which operation to apply on the cache depending upon rules implemented in Conflict Resolver. By default it is time based Conflict Resolution which means we only apply operations from bridge that are latest as compared to existing operations in cache.

- **Enable Conflict Resolver**
- **Assembly Name:**
- **Class Name:**

Specify additional configuration parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
</table>

Deploy Conflict Resolver allows to copy/update conflict resolver assemblies to the cache nodes. On adding any conflict resolver assemblies, use the ‘Deploy Conflict Resolver’ button to select the appropriate assemblies for deployment.

- Name of selected assembly will appear in **Assembly Name** textbox and all of its classes, which have implemented the **IBridgeConflictResolver** interface will be listed down in **Class Name** list box.
- Select the required class.
Conflict Resolver is used to resolve the conflict between existing operations and operation replicated from bridge to cache. Conflict arises when we try to replicate an operation from bridge for a key that already exist in cache. It decides which operation to apply on the cache depending upon rules implemented in Conflict Resolver. By default it is time based Conflict Resolution which means we only apply operations from bridge that are latest as compared to existing operations in cache.

<table>
<thead>
<tr>
<th>Assembly Name:</th>
<th>Providers, Version=1.0.0.27913, Culture=neutral, Public</th>
</tr>
</thead>
</table>

Specify additional conflict resolver classes:

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
</table>

Deploy ConflictResolver allows to copy/update conflict resolver assemblies to the cache nodes. On adding any conflict resolver assemblies, use the 'Deploy ConflictResolver' button to select the appropriate assemblies for deployment.

- Click **New** to provide any additional parameters, if required.
New Parameter dialog appears; provide the Parameter Name and Parameter Value.

Step 6 is optional. If your implementation of IBridgeConflictResolver interface does not require any initial parameters, then there is no need to provide parameter(s).

Now click the **Deploy Conflict Resolver** button, select your assembly in which you have implemented IBridgeConflictResolver interface and click on the **Open** button in Open dialog.
- Apply configuration using right clicking on the cache name in **Cache Explorer** and selecting **Apply Configuration** option.
6.6. Changing Cache Synchronization Modes (Active/Passive)

The NCache Manager provides the ability to change the synchronization mode of clustered cache connected to the bridge. This can be done by either making an active cache passive or a passive cache active.

6.6.1. Making a Passive Cache as Active Cache

NCache Manager provides the facility to convert a passive cache in a bridge to an active cache using the following steps:

Using NCache Manager
- Right click on the passive cache name under Caches tree node of a bridge in the Cache Explorer.
- Click on the Make Active option in the right click context menu.

6.6.2. Making an Active Cache as Passive Cache

NCache Manager provides the facility to convert an active cache in a bridge to a passive cache using the following steps:

Using NCache Manager
- Right click on the passive cache name under Caches tree node of a bridge in the Cache Explorer.
- Click on the Make Passive option in the right click context menu.
6.7. Configuring Master Cache

You can make an active cache (connected to a bridge) master. Please follow these steps:

**Using NCache Manager**

- Right click on the active cache name (which is not marked as master cache) under Caches tree node of a bridge in the Cache Explorer.
- Click on the Make Master option in the right click context menu.
6.8. Force State Transfer

State transfer is a process in which bridge ask the master cache to re-replicate all of its cached objects to the bridge. The main benefit of this process is to make sure the other cache in this bridge is resynchronized with the master cache. By default, state transfer is initiated on the change of master cache, cache connection after disconnection or on cluster restart.

You can execute state transfer any time after bridge starts to synchronize both caches.

Using NCache Manager

- Right click on the active cache name (which is not marked as master cache) under Caches tree node of a bridge in the Cache Explorer.
- Click on the Force State Transfer option in the right click context menu.
6.9. Connect/Disconnect a Cache

By default all added clusters are connected with bridge on start. You can disconnect then re-connect a running cluster from bridge.

Using NCache Manager

- Right click on the cache name under Caches tree node of a bridge in the Cache Explorer.
- Click on the Disconnect option in the right click context menu.

This option is only available on connected caches otherwise Connect option is provided to reconnect the disconnected clusters.
6.10. Remove a Cache from the Bridge

You can remove a cache from bridge after the bridge stops.

Using NCache Manager

- Right click on the cache name under Caches tree node of a bridge in the Cache Explorer.
- Click on the Remove option in the right click context menu.

Make sure bridge is not running otherwise it prompts to stop the bridge.
6.11. Bridge Management

You can manage a bridge using NCache Manager. It allows you to perform different management operations, e.g., add, remove etc., on a bridge. The following section states how to perform different management operations on a bridge.

6.11.1. Add Bridge Node

**Using NCache Manager**

- Right click on the bridge name under **Bridges** tree node of a bridge in the **Cache Explorer**.
- Click on the **Add Bridge Node** option in the right click menu.

- Specify the server IP address in text box or select a server from the **Network Neighborhood** listed under **Available Nodes**.
- Click **Finish**.
Right click on the name of the bridge and select *Apply Configuration*.

### 6.11.2. Apply Configuration

**Using NCache Manager**

- Right click on the bridge name under *Bridges* tree node of a bridge in the *Cache Explorer*.
- Click on the *Apply Configuration* option in the right click menu.
6.11.3. **Reset Configuration**

**Using NCache Manager**

- Right click on the bridge name under **Bridges** tree node of a bridge in the **Cache Explorer**.
- Click on the **Reset Configuration** option in the right click menu.
6.11.4. Remove a Bridge Node

Using NCache Manager

- Right click on the bridge node under **Bridges** tree node of a bridge in the **Cache Explorer**.
- Click on the **Remove** option in the right click menu.
A message box appears asking to confirm the bridge node removal. If you are sure to remove this node then click on the Yes button.

This node will be removed from the current selected bridge if you click Yes.

6.11.5. Start Bridge

You can start a bridge using the following methods:

Using NCache Manager

- Right click on the bridge name under Bridges tree node of a bridge in the Cache Explorer.
- Click on the Start option.
The color of the Bridge name’s icon and the bridge nodes’ icon changes when it is started successfully, like this:

Using Command Line Tool

- Open command prompt.
- Go to the NCache tools directory: `cd %NCHOME%/bin/tools`.
- Use `startBridge.exe`, provide required information:
  ```
  startBridge.exe demoBridge /s 20.200.21.95
  ```
- Press **ENTER**, it will start the bridge demoBridge on server 20.200.21.95.
6.11.6. Stop Bridge

You can stop a bridge using the following tools:

**Using NCache Manager**

- Right click on the bridge name under **Bridges** tree node of a bridge in the **Cache Explorer**.
- Click on the **Stop** option.

You will notice that the color of the Bridge name’s icon and bridge server nodes’ icon will change when bridge is stopped successfully.
Using Command Line Tool

- Open command prompt.
- Go the NCache tools directory: cd %NHOME%/bin/tools.
- Use stopBridge.exe, provide the required information:

  ```command
  stopBridge.exe demoBridge /s 20.200.21.95
  ```

- Press ENTER, it will stop the bridge named demoBridge on server 20.200.21.95.

6.11.7. Remove Bridge

You can remove a bridge using the following steps:

**Using NCache Manager**

- Right click on the bridge name under Bridges tree node of a bridge in the Cache Explorer.
- Click on the Remove Bridge option in the right click context menu.

  ![Cache Explorer](image)

- A dialogue box will appear. Click Yes to delete the bridge.
6.11.8. Start Bridge Service

You can start the bridge service on a bridge node using the following steps:

Using NCache Manager

- Right click on the node under **Bridge Nodes** tree node of a bridge in the **Cache Explorer**.
- Click on the **Start Bridge Service** option in the right click menu.
6.11.9. Stop Bridge Service

You can stop the bridge service on a server node using the following steps:

**Using NCache Manager**

- Right click on the node under **Bridge Nodes** tree node of a bridge in the **Cache Explorer**.
- Click on the **Stop Bridge Service** option in the right click menu.
6.11.10. Restart Bridge Service

Bridge service can be restarted (stop and then restart) a bridge node using the following steps:

**Using NCache Manager**

- Right click on the bridge name under **Bridges** tree node of a bridge in the **Cache Explorer**.
- Click on the **Restart Bridge Service** option in the right click menu.

This stops the bridge on the given server and then starts it automatically on the same server.
7. Monitoring

NCache publishes different performance counters, provides logging for caches, cache clients, and email notifications for clustered caches. Using these options, you can monitor the caches.

7.1. Cache Counters

7.1.1. Performance Counters

NCache is not only enriched with powerful monitoring facilities, but besides that; it also provides different categories of counters for windows performance monitor. This lets the clients monitor the overall performance of NCache. These categories include cluster, client, server and bridge. They cover all aspects of NCache monitoring including remote clients, cache clients, clustered operations, server counters, bridge queue status counters and many more.

Performance counters provide valuable information about system hardware, services, networks and applications that can be used to determine their performance. It helps in diagnosing problems, used to fine-tune systems and applications, monitoring real-time application performance and resolving bottleneck cause of system components.

Performance counters can be seen both from PerfMon and NCache Monitor. The statistics view is also used to see the performance statistics of a cache or a cluster.

These counters include:

<table>
<thead>
<tr>
<th>Counters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fetches/sec</td>
<td>Number of Get operations per second.</td>
</tr>
<tr>
<td>Hits/sec</td>
<td>Number of successful Get operations per second.</td>
</tr>
<tr>
<td>Misses/sec</td>
<td>Number of failed Get operations per second.</td>
</tr>
<tr>
<td>Additions/sec</td>
<td>Number of Add operations per second.</td>
</tr>
<tr>
<td>Updates/sec</td>
<td>Number of Insert operations per second.</td>
</tr>
<tr>
<td>Deletes/sec</td>
<td>Number of Remove operations per second.</td>
</tr>
<tr>
<td>Expirations/sec</td>
<td>Number of items being expired currently per second.</td>
</tr>
<tr>
<td>Evictions/sec</td>
<td>Number of items evicted per second.</td>
</tr>
<tr>
<td>Data Bal/sec</td>
<td>Number of items this node is either reading from other nodes or sending to other nodes during a Data Load Balancing mode.</td>
</tr>
<tr>
<td>Mirror Queue Size</td>
<td>Number of items in the Mirror queue.</td>
</tr>
<tr>
<td>Average sec/fetch</td>
<td>Average time in seconds taken to complete one fetch operation.</td>
</tr>
<tr>
<td>Average sec/add</td>
<td>Average time in seconds taken to complete one Add operation.</td>
</tr>
<tr>
<td>Average sec/insert</td>
<td>Average time in seconds taken to complete one insert operation.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Average sec/remove</strong></td>
<td>Average time in seconds taken to complete one remove operation.</td>
</tr>
<tr>
<td><strong>Average sec/cache operation</strong></td>
<td>Average time, in seconds, taken to complete one cache operation.</td>
</tr>
<tr>
<td><strong>Client Request/sec</strong></td>
<td>Number of requests sent by all clients to the cache server.</td>
</tr>
<tr>
<td><strong>Client Response/sec</strong></td>
<td>Number of responses received by all clients from the cache server.</td>
</tr>
<tr>
<td><strong>Clustered opsrecv/sec</strong></td>
<td>Number of clustered operations performed per second.</td>
</tr>
<tr>
<td><strong>Clustered opsent/sec</strong></td>
<td>Number of clustered operations performed per second.</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Number of items in the cache.</td>
</tr>
<tr>
<td><strong>DispatchEnter/sec</strong></td>
<td>Number of clustered operations performed per second.</td>
</tr>
<tr>
<td><strong>Mirror queue size</strong></td>
<td>Number of items in the Mirror queue.</td>
</tr>
<tr>
<td><strong>Misses/sec</strong></td>
<td>Number of failed Get operations per second.</td>
</tr>
<tr>
<td><strong>Readthru/sec</strong></td>
<td>Number of Read-through operations per second.</td>
</tr>
<tr>
<td><strong>Response sent/sec</strong></td>
<td>Number of clustered operations performed per second.</td>
</tr>
<tr>
<td><strong>Responses/sec</strong></td>
<td>Number of responses sent (meaning cache response for commands like add, get, insert, remove etc.) to all clients by this cache server.</td>
</tr>
<tr>
<td><strong>State transfer/sec</strong></td>
<td>Number of items this node is either reading from other nodes or sending to other nodes during a state transfer mode.</td>
</tr>
<tr>
<td><strong>Writethru/sec</strong></td>
<td>Number of Write-through/Write-behind operations per second.</td>
</tr>
<tr>
<td><strong>CacheLastAccessCount</strong></td>
<td>Number of items which are older than the access interval specified in the config.</td>
</tr>
<tr>
<td><strong>Cluster ops/sec</strong></td>
<td>Number of clustered operations performed per second.</td>
</tr>
<tr>
<td><strong>Bytes sent/sec</strong></td>
<td>How much data is sent from this node to other nodes in the cache.</td>
</tr>
<tr>
<td><strong>Bytes received/sec</strong></td>
<td>How much data is received by this node from other nodes in the cache.</td>
</tr>
<tr>
<td><strong>Requests/sec</strong></td>
<td>Number of requests (meaning cache commands like add, get, insert, remove etc.) being processed from all clients by this cache server.</td>
</tr>
<tr>
<td><strong>Client bytes sent/sec</strong></td>
<td>Bytes being sent from cache server to all its clients.</td>
</tr>
<tr>
<td><strong>Client bytes received/sec</strong></td>
<td>Bytes being received by cache server from all its clients.</td>
</tr>
<tr>
<td><strong>Response Queue Count</strong></td>
<td>Number of items in response queue; all responses are queued in the response queue before being sent.</td>
</tr>
<tr>
<td><strong>Response Queue Size</strong></td>
<td>Size of the response queue in bytes. Displays the total data size of all the responses in the response queue.</td>
</tr>
</tbody>
</table>
### Event Queue Count
- **Number of events in the event queue.**

### Events Processed/sec
- **Number of events processed per second on a client.**

### Events Triggered/sec
- **Number of events triggered and received by client per second.**

### Average ms/event
- **Average time taken by client to process a single event.**

### MapReduce Running Tasks
- **Number of tasks are in execution state at a time.**

### MapReduce Waiting Tasks
- **Number of tasks in waiting state when the maximum running tasks limit for execution has been reached.**

### MapReduce Mapped/sec
- **Number of keys mapped per second in running tasks.**

### MapReduce Reduced/sec
- **Number of keys reduced per second in running tasks.**

### MapReduce Combined/sec
- **Number of keys combined per second in running tasks.**

### Average µs/addBulk
- **Average Number of bulk (not independent keys) adds per microsecond. This applies to Bulk, Group and Tag APIs.**

### Average µs/fetchBulk
- **Average Number of bulk (not independent keys) fetches per microsecond. This applies to Bulk, Group and Tag APIs.**

### Average µs/removeBulk
- **Average Number of bulk (not independent keys) removes per microsecond. This applies to Bulk, Group and Tag APIs.**

### Average µs/insertBulk
- **Average Number of bulk (not independent keys) inserts per microsecond. This applies to Bulk, Group and Tag APIs.**

#### 7.1.2. Debugging Counters

<table>
<thead>
<tr>
<th>Counter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socket send time (ms)</td>
<td>Time in milliseconds taken to send a request.</td>
</tr>
<tr>
<td>Socket send size (bytes)</td>
<td>Bytes sent per second.</td>
</tr>
<tr>
<td>Socket recv time (ms)</td>
<td>Time in milliseconds taken to receive a request.</td>
</tr>
<tr>
<td>Socket recv size (bytes)</td>
<td>Bytes received per second.</td>
</tr>
<tr>
<td>Nagling msg count</td>
<td>How many messages were sent together in socket.send.</td>
</tr>
<tr>
<td>TcpUpQueueCount</td>
<td>Number of items in TCP up queue.</td>
</tr>
<tr>
<td>TcpDownQueueCount</td>
<td>Number of items in TCP down queue.</td>
</tr>
<tr>
<td>BCastQueueCount</td>
<td>Number of items in BCast Queue.</td>
</tr>
<tr>
<td>MCastQueueCount</td>
<td>Number of items in MCast Queue.</td>
</tr>
</tbody>
</table>
7.2. Monitoring Caches Using NCache Manager

NCache Manager is a tool to manage the caches and bridge, but it also provides the basic monitoring for the caches. In this way, you are able to manage and monitor your caches at a single place.

Using NCache Manager, you can also configure the different monitoring settings. The following sections provide an overview of how to configure these settings.

7.2.1. Changing the Management Port of a Cache Node

All performance and connectivity related data is fetched using a particular port called NCache Management Port. Default value of this port is 8250 but it is configurable using NCache Manager.

Using NCache Manager

- Go to Tools menu of NCache Manager.

- Select Options. A dialog box appears with Statistics, Management and Startup tabs.

- Go to the Management tab.
• Type any value within your system’s port range and make sure that the port is free. Click OK.

You can change this value back to default any time using Restore Defaults button available on this tab.
7.2.2. Configuring Counters to Display

By Default, a few important counters are displayed in statistics window but you can add/remove more counters according to your requirements.

Using NCache Manager

- Go to Tools menu of NCache Manager.

- Select Options. A dialog box appears with Statistics, Management and Startup tabs.

- Go to the Statistics tab.
• All performance counters are displayed in this tab in categorical view; you can select/deselect counters from this tab.

• Click OK to save your selection.
7.1.4. Cluster Health

You can view cluster health using NCache Manager. Cluster health shows the connection status of each server in cache cluster. There are three types of Status:

- **Fully Connected**: The cache is running and all server nodes are connected with each other.
- **Partially Connected**: The cache is running but not all server nodes are connected with each other.
- **Cache Stopped**: The cache is running but the server with this status is stopped.

**Using NCache Manager**

- Right click on the cache name under **Clustered Caches** tree node in **Cache Explorer** view to bring up the context menu.
  
  ![Cache Explorer](image)

- Select View Cluster Connectivity option. Cluster Connectivity dialog box appears.
This screen shot shows the cluster connectivity of a two node clustered cache, one server is running and the other is stopped.

Click Close to dismiss the dialog.
7.1.5. Monitoring Clustered Caches Using NCache Email Alerts

NCache also provides the Email Alerts (notifications) for different cache activities. You can choose the required (few or all) events for which you want to receive the email alerts from NCache, so that you remain fully aware what is going on. And in case of any serious issue, you can manage the cache accordingly.

- Configure the Email Alerts for a cache using NCache Manager.
- Stop the cache for which you have configured email alerts.
- Now check your email, which you have provided for receiving email from NCache. You will see the an email from NCache with contents like this:

```plaintext
Machine: TEST_VM1
Cache Name: demoCache
Event Type: Cache Stop
Event ID: 1001
Date Time: 4/11/2014 4:38:01 PM

Message: "demoCache" stopped successfully.

NCache Alerts
PS: This is NCache auto generated email.
```
7.2. Monitoring Caches Using NCache Monitor

Clustered caches can be monitored to view performance counters in live visual graphs using NCache Monitor shipped with NCache Enterprise edition. In NCache Monitor, we have two types of dashboards; Graphical and Report View

- **Graphical View**
In graphical dashboards, counters are shown in the form of graphs with respect to time. Counter value will be plotted along Y-Axis and time on X-Axis. Graphical dashboards are helpful in the scenarios where you want to monitor cache for some specific time.

- **Report View**
In report view dashboards, counters are shown in the form of numeric values. Report view dashboards are helpful in the scenarios when you want to monitor instantaneous performance of NCache.

7.2.1. Monitoring Cache Server Counters Using Built-In Dashboard

There are two major categories of performance counters; **Server counters** and **Client counters**.

**Launch NCache Monitor using NCache Manager**

- Right click on the clustered cache name under **Clustered Cached** tree node in **Cache Explorer** context menu.

- Select **Monitor Cluster** option from the right click menu.
- NCache Monitor is opened. By default there are two dashboards already opened, **Server Dashboard** and **Report-View Dashboard**.

Both default dashboards are non-editable. It means no counter or graph can be added or removed from default dashboards.

**Server Dashboard**

It contains **Cluster Health**, **Event Logs** along with some mostly required cache counter graphs like, **NCache CPU Graph**, **NCache Network Graph**, **Cache Size**, **NCache Request/sec Graph** and **NCache Memory Graph** etc. Server Dashboard is generally a graphical view dashboard.
Report-View Dashboard

This view shows counters in numeric form rather than graphs. It contains two portions, each for **Server Report View** and **Client Report View**. Both portions contain respective counters.

### Server Report View

<table>
<thead>
<tr>
<th>Counters</th>
<th>20.200.21.144</th>
<th>20.200.21.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Usage</td>
<td>40.000</td>
<td>58.000</td>
</tr>
<tr>
<td>Memory Usage</td>
<td>1411.672</td>
<td>1472.297</td>
</tr>
<tr>
<td>Network Usage</td>
<td>0.431</td>
<td>0.414</td>
</tr>
<tr>
<td>Cache Size (MB)</td>
<td>110.843</td>
<td>130.175</td>
</tr>
<tr>
<td>Request/sec</td>
<td>2423.816</td>
<td>2250.159</td>
</tr>
<tr>
<td>Additions/sec</td>
<td>1202.616</td>
<td>1129.071</td>
</tr>
<tr>
<td>Fetches/sec</td>
<td>1207.301</td>
<td>1126.617</td>
</tr>
<tr>
<td>Updates/sec</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Deletes/sec</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Item Count</td>
<td>94945.000</td>
<td>111519.000</td>
</tr>
<tr>
<td>System CPU Usage</td>
<td>46.514</td>
<td>66.154</td>
</tr>
<tr>
<td>System Memory Usage</td>
<td>1937.113</td>
<td>1933.043</td>
</tr>
</tbody>
</table>

### Client Report View

<table>
<thead>
<tr>
<th>Counters</th>
<th>20.200.21.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Usage</td>
<td>14.000</td>
</tr>
<tr>
<td>Memory Usage</td>
<td>69.000</td>
</tr>
<tr>
<td>Network Usage</td>
<td>---</td>
</tr>
<tr>
<td>Read Operations/sec</td>
<td>900.000</td>
</tr>
<tr>
<td>Write Operations/sec</td>
<td>900.000</td>
</tr>
<tr>
<td>Addition/sec</td>
<td>0.000</td>
</tr>
<tr>
<td>Deletes/sec</td>
<td>0.000</td>
</tr>
<tr>
<td>Fetches/sec</td>
<td>545.000</td>
</tr>
<tr>
<td>Updates/sec</td>
<td>546.000</td>
</tr>
<tr>
<td>Request Queue size</td>
<td>2.000</td>
</tr>
</tbody>
</table>
7.2.2. Monitoring Cache Client Counters Using Built-In Client Dashboard

NCache Monitor also provides built-in dashboard named as **Client Dashboard**. This dashboard has all required counters/graph in it for the clients' performance monitoring. It can be viewed for monitoring as follows:

- Click on the **View** menu of the **NCache Monitor** and click on the **Client Dashboard** menu option.

  ![Client Dashboard menu]

- **Client Dashboard** appears and it consists of different client side graphs.

  ![Client Dashboard]

- This built-in dashboard displays different client side counters. Few of the counters display the NCache client system resource usage info like **NCache Client CPU**, **NCache Client Memory**, etc.
**NCache Client Network.** This dashboard also shows the NCache client specific information, e.g., NCache Client Request Queue, NCache Client Read Operations/sec, NCache Client Write Operations.

- It also shows the ASP.NET session counters like ASP.NET Requests/sec, ASP.NET Pending Queue/sec, ASP.NET Total Sessions.

### 7.2.3. Monitoring Clustered Cache Counters using Custom Dashboard

You can also add a custom dashboard to place the required counters on it for the monitoring using NCache Monitor.

- Click on the **File** menu of the **NCache Monitor**.
- Choose the **New Dashboard** menu option.

**OR**

- Click on the **+** button on the Toolbar.
- **Create Dashboard Wizard** appears. Provide **Dashboard Name** and select the **Dashboard Layout** (Graphical or Report View).
- Click **Next**.

- Select the number of rows and column for this dashboard and click on the **Finish** button.
- New dashboard **Dashboard1** is created with specified number of rows and columns.

- Now drag and drop the required counters from **Cache Server** category or from **Cache Client** category that exists in the **Toolbox** available on the left side of NCache Monitor.
The custom dashboard after placing different server and client counters on it will look similar to this:
You can override the already placed counters in custom dashboard with new ones. Just drag and drop the new counter from the Toolbox on a current counter/graph and the new counter/graph replaces the previous one.

You can also close/remove a counter from custom dashboard by clicking on the red cross button available on the top right corner of every counter/graph.
7.2.4. Monitoring PerfMon Counters Using Custom Dashboard

Performance Monitor is a Microsoft Windows tool that can be used within NCache Monitor custom dashboards just like other NCache custom controls. Follow the steps given below to add Perfmon control in a custom dashboard:

- Go to the Toolbox, available on the left side inside NCache Monitor.
- Click on the Cache Server button on the Toolbox.
- Select the Perfmon control; drag and drop it in any cell inside a custom dashboard.

- Now you can add the required NCache counters in it or any other Perfmon based counters. Click on the green plus button, select the required server machine and then choose the required Perfmon counters.

- You can find details about Perfmon tool and its usage on Using Performance Monitor.
7.2.5. Monitoring Cache Event Logs Using Custom Dashboard

Event Logs control displays log entries specific to cache. NCache logs multiple events’ information (Node left, Node Joins, Cache Started, Cache Stopped etc.) in windows event logs. Using this control, you can monitor all of the cache events in a single place. Details of these events along with date time can be monitored using Event Logs control.

<table>
<thead>
<tr>
<th>Type</th>
<th>Node</th>
<th>Date</th>
<th>Time</th>
<th>Event</th>
<th>Source</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>20.200.21.95</td>
<td>6/9/2014</td>
<td>05:32:3...</td>
<td>1008</td>
<td>NCache</td>
<td>&quot;democlusteredcache(20.200.21.95:8820)&quot; has ended state transfer.</td>
</tr>
</tbody>
</table>

- Double click on any log entry to open it with full details in new window, like this:

```
Event Details

Event:
Date: 6/9/2014  Source: NCache
Time: 17:32:41  Type: Information
Event ID: 1009  Computer: 20.200.21.95

Description:
"democlusteredcache(20.200.21.95:8820)" has ended state transfer.
```

© Copyright 2015 Alachisoft
You can also add any other event source, which publishes events in windows event logs using the NCache Monitor ‘Event logs’ control. Please follow the steps given below to add any other event source for monitoring in NCache Monitor Event Log control:

- Click on the **Tools** -> **Options** in NCache Monitor.

- A dialog box will open up. Click on the **Event Log** option in the left side menu.

- Specify the name of the **Event source** (any application or service that publishes events logs).

- Click **Add**.
You can also remove any of the added event source for event monitoring. Click on the Remove button against the event source which you want to remove.

- Click OK to save the changes. Now NCache Monitor starts event log monitoring for this newly added event source too (along with all of the previously added event sources).

7.2.6. Monitoring Client Process of a Clustered Cache Using Custom Dashboard

**Client Process** control shows details of running client applications against current cache. Details include client IP, process ID, port, server IP with which this client application is connected and Sent/Received bytes by each client.

Follow the steps given below to add **Client Process** control in a custom dashboard:

- Go to the **Toolbox**, available on the left side inside NCache Monitor.
- Click on the **Cache Server** button on the **Toolbox**.
- Select the **Client Process** control; drag and drop it into any cell inside a custom dashboard.
Information of all the currently connected clients displayed in Client Process control for current cache.

Client processes can be grouped using any column. Drag and drop a column header on Drag a column header here to group by that column. Below image is an example of grouping by Server IP field.

### 7.2.7. ASP.NET Session Monitoring in Custom Dashboard

Using NCache Monitor you can also monitor the ASP.NET session counters. Currently you can monitor the ASP.NET Requests/sec, ASP.NET Pending Requests and ASP.NET Total Session counter in NCache Monitor. Controls for these counters exist in Toolbox under Cache Client category.
Just drag and drop these controls in a custom dashboard, and NCache Monitor starts fetching the values of these counters.
7.3. Monitoring Bridge Using NCache Manager

Bridge is used to synchronize caches across the WAN. Bridge can be monitored using NCache Manager and NCache Monitor.

7.3.1. Configuring Bridge Counters to Display

By Default, a few important counters are displayed in statistics window but you can add/remove cluster counters connected to bridge according to your requirements.

Using NCache Manager

- Go to Tools menu of NCache Manager.

- Select Options menu and a dialog box appears with Statistics, Management and Startup tabs. Go to the Statistics tab.
- Check the Show counters of Caches connected with bridge checkbox.
- Click OK.
7.3.2. Browse Bridge Statistics

Using NCache Manager

- Right click on the bridge name under the **Bridges** tree node in **Cache Explorer** to bring up context menu.

- Click on the **Statistics** option. It opens statistics tab for this bridge in the right pane.
7.4. Logging

7.4.1. NCache Log Viewer

While log files are one of the essentials in any processing, their readability and organization is always overlooked, making the task tedious of finding the problem. NCache Log Viewer is an interactive GUI tool dedicated completely to logging, as it categorizes your log entries into separate fields and lets you customize and search entries in a much more convenient manner. You can also view multiple log files for comparison.

The features provided by NCache Log Viewer to enhance your readability are as follows:

**Open Files**

- Click on the button or type **CTRL+O** to open the files you want to import into NCache Log Viewer.
After selecting the log file, click **Open**.

The logs are imported into NCache Log Viewer, showing the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>The cache node on which the operations have taken place.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>The timestamp of each log, in the format <code>yyyy-mm-dd h:mm:ss, ms</code>.</td>
</tr>
<tr>
<td>AppDomain</td>
<td>The process name.</td>
</tr>
<tr>
<td>LoggerName</td>
<td>Name of the component of which the information is logged.</td>
</tr>
<tr>
<td>ThreadName</td>
<td>Identification of thread</td>
</tr>
<tr>
<td>Level</td>
<td>Level shows the type of log, either it is INFO or ERROR</td>
</tr>
<tr>
<td>Message</td>
<td>The detailed log information about success/failure of operations, nodes joining or leaving, services starting or stopping.</td>
</tr>
</tbody>
</table>
In order to view the Message of the log without scrolling, click on any log.

The message is displayed in the **Message** division at the bottom of the Log Viewer.

- Click on the button or type **CTRL+S** to export/save the file to any location.
• After selecting the location, click **Save**.

**Search**

• To search for a certain term or option in the viewer, select the option from the **Any Field** drop down menu on the toolbar.

• If you want to find the term in the whole log file and not specify the parameters, select **Any Field**.

• In the text bar next to the Any Field drop down menu, enter the term you want to search. It displays suggestions containing that term in a drop down menu, easing your search process.
After entering the option, click on the **button.

If you have selected an option from the drop down menu, press **Enter** or click on the **button.

**Split Logs based on Field**

- You can also view the logs by splitting the entries based on one or more unique field entries using the **button next to each field name. For example, in the ThreadName field, there are 3 unique threads in the sample log:

- Selecting one of the options will display the logs for that ThreadName only:

- In order to clear the split log view, click on the **button. It will now be displaying the option of **Clear**.
Reload

- If changes have been made to your cache or service, it is advisable to reload the log file to view the updated changes.
- To reload the updated log file, click

Clear All Logs

- In order to clear the logs from the viewer, click on the button.

Toggle Bookmark on log entry

- In case you want to bookmark a log entry for future reference, click on the entry.
- The button becomes active.
- Click on the active button to bookmark the entry.
- It marks a bookmark icon next to the entry.

Bookmark Navigation

- Once an entry has been bookmarked, the and buttons become active.
- You can navigate from one bookmark to another using the Previous Bookmark and Next Bookmark button.

Settings

- You can customize the fields and number of entries displayed in the viewer according to your preference using the button on the toolbar.
1. **Minimize to Tray**
Next time you minimize the log viewer, it will be minimized in the tray. You can click on the tray icon to maximize the Log Viewer.

2. **Show multiple node files in colors**
If you add multiple log files of different nodes, you can check this option to differentiate between the nodes and increase readability.
3. **Max log entries in live mode**
   You can customize the maximum number of entries to be shown in the viewer while it is running.

4. **Columns to View**
   You can customize the columns you want to view or remove from the viewer by checking/unchecking the options.

**Note:** Some settings will take effect after NCache Log Viewer is restarted.
7.4.2. NCache Monitor Logging

NCache Monitor also provides the logging functionality for different server and client counters with the help of Report View dashboard. Server counters of ‘Server Report View’ and Client counters of ‘Client Report View’ can be logged independently.

NCache Monitor Report view also provides the functionality to schedule logging, where the start and stop timings of logging counters can be specified.

Run-time statistics of all counters specified in the specific report view panel, are logged in files of .CSV format. Default path of these files is different for server and client counters as:

- **Server Report View** panel counters are logged in:
  
  %InstalledDirectory%\log-files\NCacheMonitor\Server\ 

- **Client Report View** panel counters are logged in:
  
  %InstalledDirectory%\log-files\NCacheMonitor\Client\ 

This logged data can be used for later evaluation of cache cluster and clients’ performance and health.
7.4.3. Windows Event Logging

NCache provides support to view events in Windows Event Log. NCache logs its important events in Window Event Logs. This allows monitoring all such events through Windows Event Viewer or with any other tool that supports Window Event Logger. Some of the events that are logged include:

- All NCache installation events. These events are combination of successful or aborted installation.
- All errors encountered by NCache, Bridge and Memcached services.
- Service start and stop events
- Cache start and stop
- Cache server joining or leaving a cache cluster
- Cache size going above a configurable 80% threshold

![Windows Event Log Example](image)

7.4.4. Performance Counters Logging

Using Report-View you can instantly monitor performance but you can also log these counters in a CSV file. All the counter values, which are currently in dashboard will be logged into a .CSV file after each second. You can start the performance counters logging instantly or you can schedule the logging on a specific time.

**Manual Logging**

Follow the below given steps to log the counters using NCache Monitor:

- Go to the **Report-View Dashboard** (built-in Report-View dashboard or your created dashboard).
- Click on the **Start Logging** button icon existing on the report view control. This starts the logging.
Log file can be found under `%NCHOME%/log-files/NCacheMonitor`

You can stop the performance counter logging by clicking on the **Stop Logging** icon in the Report-View Dashboard.

### Schedule Logging

Rather than manually starting and stopping logging you can also schedule logging. This way you can obtain logging data for a particular period of time.

- Click on the **Schedule Logging** icon in Report-View dashboard.
7.5. Monitoring NCache Using Windows PerfmonTool

NCache publishes its performance counters through Perfmon. You can monitor NCache counters for a specific cache on any Windows machine using Windows Perfmon tool.

7.5.1. Monitoring Cache Server Counters Using PerfmonTool

NCache publishes cache server counters in Perfmon under category NCache. This category has all the counters related to the cache server. Follow the steps given below to monitor the NCache counters through Perfmon tool:

- Press WINDOWS + R keys on your keyboard or click on the Windows start menu and then type Perfmon and press ENTER key.

- Perfmon tool opens up. Click on the Performance Monitor under Monitoring Tools.
Click on the cross (X) button to remove the default counter which is already added in it. Then click on the plus (+) button to open the Add Counters dialog box.

Using the vertical slider of available counters list box, scroll upward to find **NCache** category.

Click on the down arrowhead icon to expand the **NCache** category. All of its counters are listed under it. Select the required counters from this list like this:
- All of the current running caches (and replicas of caches) appear inside of **Instances of selected objects** list box.

- Select the required instance or simply click on the **<All instances>** and click on the **Add >>** button.

- All of the selected counters (selected in step 5) for all the selected instances of caches appear in **Added counters** list box on the right side.
Click OK. All of the selected counters will appear in Perfmon tool like this:
You can also change the view of Perfmon from **Report** view to **Line** view. Follow the steps given below:

1. Click on the Line option of the drop down menu. This changes the Perfmon view from **Report** to **Line**. In Line view, counters values are displayed in the form of continuous graph, like this:
7.5.2. Monitoring Cache Client Counters Using Perfmon Tool

NCache publishes cache client counters in Perfmon under the category **NCache Client**. This category has all the counters related to the cache client. Follow these steps to monitor the NCache client side counters through Perfmon tool:

- Press **WINDOWS + R** keys on your keyboard OR click on the windows start menu and then type Perfmon and press **ENTER** key.

- Perfmon tool opens up; Click on the **Performance Monitor** under **Monitoring Tools**.

- Click on the cross (X) button to remove the default counter which is already added in it. Then click on the plus (+) button to open the **Add Counters** dialog box.
Using the vertical slider of available counters list box, scroll upward to find **NCache client** category.
Click on the down arrowhead icon to expand the **NCache Client** category. All of its counters are listed under it. Select the required counters from this list like this:
All of the current running caches for which clients are running appear inside of **Instances of selected objects** list box.

Select the required instance or simply click on the **<All instances>**, and click on the **Add >>** button.

All of the selected counters (selected in step 5) for all the selected instances of caches appear in **Added counters** list box (exist on the right side).

Click **OK**. All of the selected counters will appear in Perfmon tool like this:
7.5.3. Monitoring Bridge Counters Using Perfmon Tool

NCache publishes bridge counters in Perfmon under category **NCache Bridge**. This category has all the counters related to the bridge. Follow these steps to monitor the NCache bridge counters through Perfmon tool:

- Press **WINDOWS + R** keys on your keyboard OR click on the windows start menu and then type **Perfmon** and press **ENTER** key.
- Perfmon tool opens up; click on the **Performance Monitor** under **Monitoring Tools**.
- Click on the cross (X) button to remove the default counter which is already added in it. Then click on the plus (+) button to open the **Add Counters** dialog box.
- Using the vertical slider of available counters list box, scroll upward to find **NCache Bridge** category.
- Click on the down arrowhead (icon) to expand the **NCache Bridge** category. All of its counters are listed under it. Select the required counters from this list like this:
- All of the current running caches for which clients are running appear inside of the **Instances of selected objects** list box.

- Select the required instance or simply click on the `<All instances>`, and click on the **Add >>** button.

- All of the counters that were selected in step 5 for all the selected instance of caches appear in the **Added counters** list box on the right side.
Click **OK** available at the bottom right of this dialog box. All of the selected counters will appear in Perfmon tool like this:
8. IPV6 Support

NCache supports IPV6 addresses for both cluster wide and client-server communication. All that the user needs to do is assigning IPV6 addresses to tags BindToClusterIP and BindToClientServerIP in Alachisoft.NCache.Service.exe.config.

```xml
<add key="NCacheServer.BindToClusterIP" value="FE80::0202:B3FF:FE1E:8329"/>

<add key="NCacheServer.BindToClientServerIP" value="FE80::0202:B3FF:FE1E:8329"/>
```

However the only limitation at present is that cache statistics in NCache Manager cannot be fetched using IPV6. This is because NCache Manager employs a third party tool to fetch and show cache statistics using Perfmon. This tool, at the moment, does not support IPV6 addresses. Therefore, it is recommended to use host name (server name) instead for fetching cache statistics if NCache cluster is on IPV6.

In NCache Command Line tools where server name or IPV6 address is required as argument, it is recommended to use IPV6 address instead of server name. This is because using server name might run the tool on such an IP address on server where NCache service is not running. This will result in an exception.