

***Alachisoft***

**NCache 4.1 SP1**

---

**NCache Release Notes  
Bugs Fixes and Enhancements**

**Last Modified:**

**May 16, 2012**

## Table of Contents

Release Notes NCache 4.1 (Service Pack 1) .....	3
Release Notes NCache 4.1 .....	5
Release Notes NCache 3.8 (Service Pack 4) .....	8
Release Notes NCache 3.8 (Service Pack 3) .....	10
Release Notes NCache 3.8 (Service Pack 2) .....	11
Release Notes NCache 3.8 (Service Pack 1) .....	13
Release Notes NCache 3.8 .....	15
Release Notes NCache 3.6.2 (Service Pack 3) .....	17
Release Notes NCache 3.6.2 (Service Pack 2) .....	19
Release Notes NCache 3.6.2 (Service Pack 1) .....	21
Release Notes NCache 3.6.2 .....	23
Release Notes NCache 3.6.1 .....	24

## Release Notes NCache 4.1 (Service Pack 1)

Wednesday, May 16, 2012

### Introduction

NCache 4.1 Service Pack1 (SP1) contains some important enhancements and major bug fixes relating to the stability of the cache. It is recommended upgrade for all NCache 4.1 users.

### Enhancements and New additions

Following are some enhancements made in this release.

**1. Generics types support for Compact Serialization**

You can add custom generic types for compact serialization. All generic types with any number of arguments can be serialized through compact serialization. You can register generic types through NCache Manager or through a custom handler by implementing the interface *IGenericTypes*. Currently, this feature is only available for .NET clients.

**2. Update CacheItem hint at runtime**

You can now modify cache item attributes at runtime without modifying the data. Currently, the API (*SetAttribute*) allows you to update dependency and expiration hints.

**3. UTC support for Different Time zones**

You can have cache servers as well as clients running under different time zones; NCache will maintain a standard time to expire an item on the basis of local time zone. Whenever an item replicates or moves from one cache server to the other, the expiration is reset according to the local time zone. This makes the item expiration possible according to the configured timeslot.

### List of Bugs Fixed:

[741101](#) FIX: Client was unable to initialize NCache through a SQL CLR based procedure or trigger.

[741102](#) FIX: Object ref exception if only WritBehind is configured for a cache and WithThru operation is performed in API.

[741103](#) FIX: Output caching in 2-node web-farm, item is deleted from cache if 2nd request goes to 2nd node of the farm.

[741104](#) FIX: Query parser throws buffer overflow if the query length exceeds 256 characters.

[741105](#) FIX: GetGroupsKeys, GetBulk, Tags, and Object Query does not return correct results during state transfer.

[741106](#) FIX: In case of partition topology only one server was sending general notifications to remote clients.

741107 FIX: NamedTags query throws exception "Index not defined" if only one item is added in partitioned cache.

741108 FIX: Java client throws exception that it is unable to read client.nconf file even though you have provided the cache-id, server-ip and port through the API.

741109 FIX: Slow response on right clicking a cache server.

741110 FIX: Content optimization log file was not generated.

741111 FIX: Licensing does not work on machines with 32 or more cores.

741112 FIX: AddDependency() resets the existing dependency instead of appending a new dependency.

741113 FIX: Client Application does not receive updated hash-map when a network is disabled on the cache server.

741114 FIX: Security was using case sensitive username.

741115 FIX: Cluster becomes partial and sometimes unresponsive as well. This happens when machine is rebooted and cache from that server tries to join the cluster again before cluster has removed this server from the its cluster membership.

## Release Notes NCache 4.1

Tuesday, August 23, 2011

### Introduction

NCache 4.1 has introduced very important new features and enhancements that are critical for enterprise level applications. This new release gives a whole new package for run time data sharing between multiple platforms (java & .Net). Java has been made 100% compatible with .NET and now you can even plug-in your Java code with NCache process. NCache request-response model has also been enhanced to handle large responses where GBs of data can be retrieved from the clustered cache in a single request. All the bugs that were reported in earlier release have been fixed in this release.

### Enhancements and New additions

For a comprehensive list of ALL FEATURES in 4.1, please read [NCache Features](#):

#### 1. Runtime Data Sharing between .NET & Java

NCache now allows you to store either .NET objects in the cache and read them as Java objects from your Java applications, or vice versa. And, instead of doing .NET to Java transformation through XML, NCache uses binary-level transformation. As a result, the performance is super fast. NCache automatically resolves type conflicts between Java and .NET.

You can also utilize multiple versions in Runtime Data Sharing between .NET and Java. See details below.

#### 2. Multiple Objects Version Support (.NET & Java)

You can now share multiple versions of the same .NET or Java classes across multiple applications. One application may be using version 1.0 of a class while another application may have a later version 2.0 of this same class. When version 2.0 of this class is stored in the cache, the earlier application can still fetch this class as version 1.0, and vice versa. NCache lets you configure version mappings through XML configuration files.

You can utilize version also in Runtime Data Sharing between .NET and Java.

#### 3. Continuous Query (.NET & Java)

NCache lets you specify a data-set based on an SQL-like query. It then maintains this data-set in the cache cluster for you and monitors any runtime changes in it, including additions, updates, or deletes. And, NCache notifies your application whenever any of these changes occur in the dataset. This allows your applications to more intelligently monitor for either data changes or addition of data matching a certain criteria and be notified by NCache.

This is a powerful feature if you want to share data at runtime between multiple applications.

#### 4. Much Enhanced Bridge Topology

Bridge Topology allows you to intelligently and asynchronously replicate the entire cache across the WAN. NCache 4.1 now offer four different configurations in Bridge Topology. They are:

1. **Active/Passive:** In this configuration, one cache is active and the other passive. Whatever updates you make to the active cache are asynchronously

applied to the passive cache by the Bridge. Since the connection between active and passive is across the WAN, it is likely to be unstable. But, the Bridge is aware of it and if the connection breaks, it automatically reestablishes it.

2. **Active/Active:** In this configuration, both caches are active and the Bridge receives update requests from both sides. Both caches are also maintaining an identical "clock" for time-stamping. This clock is synchronized through the Bridge. Whenever there is a conflict, meaning the same cache item is being updated in both caches, it is resolved by default on a last-update-wins strategy. But, if you want, you can provide a custom resolution handler which is invoked in case of a conflict. You can then determine which update should stay and which one should be discarded.
  3. **Hub-Spoke:** In this configuration, there is one active Hub and multiple passive Spokes (or satellite caches). All updates originate from the Hub and are asynchronously applied to all the Spokes. The Spokes are always read-only.
  4. **Star:** This configuration looks identical to a Hub-Spoke except that all the Spokes are also active. In this configuration, there is one centralized cache and multiple satellite active caches. All updates are synchronized through the centralized cache to ensure consistency.
5. **Named Tags (.NET & Java)**  
Previously, you could only assign tags as values. Now, you can assign tags with names. This allows you to index data based on attribute name and attribute value concept. Previously, you could index objects but all string data could not be indexed. Now, even string data (e.g. XML) could be indexed with named tags. Then, you could either use NCache API to fetch data belonging to one or more named tags or you could issue SQL-like query (through LINQ or OQL) for it.
6. **Java Features Now 100% Equivalent to .NET**  
NCache 4.1 brings support for Java at the same level as .NET. Here are the new Java based features intended to catch up to .NET feature-set:
- **Java Client API 100% equivalent to .NET**
  - **Java Client Cache (InProc):** You can now use Client Cache feature in Java applications on Windows or Unix. There is no code change required in your applications to enable Client Cache.
  - **Read-Thru, Write-Thru, Write-Behind in Java:** You can now write your Read-Thru and Write-Thru handlers in Java and register them with NCache. NCache runs your native Java code on the cache servers just like it does with .NET code.
  - **Cache Loader in Java:** Write your cache loading code in Java and register it with NCache. Your native Java code will run on cache servers just like the .NET code.
  - **Dynamic Compact Serialization for Java:** Now, you can register your Java classes with NCache and NCache generates their serialization code at initialization time, compiles it in-memory, and sends it to NCache clients. This compiled Java code is then run to serialize/de-serialize your Java classes. This obviously speeds up your performance because compiled serialization code runs much faster than Reflection based serialization that is otherwise done in Java.
  - **Spring/Hibernate Support:** NCache now provides an L2 Cache for Hibernate. This allows you to start using NCache without making any code changes in your Java application.

## 7. Multi-Response Model

NCache now allows the cache servers to return larger response in smaller chunks for a given request. This improves the overall cache performance because the serialization cost reduces with the size and also helps applications to fetch larger data sets in a single call. It also eliminates the .NET serialization limitation where it fails to serialize data size larger than 1.9 GB. Response threshold and chunk size both are configurable and can be modified from NCache service config file "NCache\bin\service\Alachisoft.NCache.Service.exe.config"

```
<!-- Response size in MB -->
<add key="NCacheServer.ResponseDataSize" value="1024"/>
```

### List of Bugs Fixed:

741001 FIX: When using Client-Cache GetCacheItem returns incorrect Group name.

741002 FIX: Object Query does not work if the IN operator has just one parameter e.g  
 Select Northwind.Customers where this.City In(?).

741003 FIX: Alachisoft.NCache.Web.dll is missing in "4.0 GAC" folder in 64 bit remote client installation of NCache

741004 FIX: Alachisoft.NCache.Security.dll (32 bit) missing in Developer installation.

741005 FIX: When a node joins the cluster the State transfer throws an object reference not set exception in cache server log file. This happens if one write through provider is configured with write behind enabled.

741006 FIX: In a two or three node partitioned replica cluster, if a node goes down and immediately comes back before the state transfer is completed, items are lost.

741007 FIX: If the indexed items are also associated with tags, the LINQ query without any where clause may return an exception that the key is already added in the dictionary. This was happening because of duplicate keys were added in the result.

741008 FIX: Cannot increase the cache size at runtime from NCache Manager.

741009 FIX: Cache server sends multiple status inquiry requests to the other cache server if it does not receive a response in a certain time for a given request. In case of network failures or cache node unresponsiveness the request queue becomes full and timeout errors are thrown to the user. This status inquiry mechanism is now disabled by default and can be turned on through NCache service config file.

741010 FIX: If there are two or more caches running on the same machines and both are using TAG feature then you may get object reference not set errors in the cache server log files. This may also slow down the cache response in some situations. This has been fixed.

741011 FIX: If the response size is larger than 1.9 GB the cache will hang and clients will timeout. This happens because of the .NET serialization limitation. This happens usually with bulk operations like Queries, Tags, Groups, and bulk operations.

## Release Notes NCache 3.8 (Service Pack 4)

Thursday, Mar 17, 2011

### Introduction

NCache 3.8 Service Pack 4 (SP4) contains important fixes which are important for production environments. It is a recommended upgrade for all NCache 3.8 users.

### Enhancements and New additions

Following are some enhancements made in this release.

#### 4. String sharing between Java and .NET (vice versa)

You can share string values between .NET and Java applications.

#### 5. SQL Cache Dependency Custom Queue/Notification Service

SQL Cache dependency architecture has been enhanced to support custom queues and notification services that does not require extra user permissions like "create queue" and "create service". NCache now allows you to choose either default mode (with default SQL notification and queue service) or custom mode where you can specify your own queue and notification service.

The service name format should be "NCacheSQLService-[ip-address]" and "NCacheSQLQueue-[ip-address]" where the IP-Address will be of machine on which NCache service process will be running. You can specify this setting in service configuration file "NCache/bin/service/Alachisoft.NCache.Service.exe.config".

```
<add key="NCacheServer.NCacheSQLNotificationService"  
value="NCacheSQLService" />
```

### List of Bugs Fixed:

[738401](#) FIX: Enumeration returns byte array if the item was added through cache loader.

[738402](#) FIX: ViewState module throws Null Reference Exception if the traces are disabled and module tries to write something in log.

[738403](#) FIX: Command line activation throws invalid license key.

[738404](#) FIX: Cache becomes unresponsive or timeouts when the number of items exceed 7 million.

[738405](#) FIX: Query Sample in x64 has some compilation errors.

[738406](#) FIX: NCache Session State Management assembly was missing in 64-bit developer installation.

[738407](#) FIX: Connection Balancing and failover in Partitioned and Partitioned Replica topologies does not work if dual NICs are used where one NIC is bound to client-server communication and the second NIC is bound to cache-server (cluster) communication.

[738408](#) FIX: A 32 bit security assembly "Alachisoft.NCache.security.dll" is missing in windows GAC in NCache 64 bit setup. This does not allow 32-bit applications to run on 64 bit caching server.

**738409** FIX: CacheProvider 4.0 - GetValues() method returns all key-values including those which are also not available in cache. As per Microsoft documentation it should return only the key/values that exist in cache.

**738410** FIX: CacheProvider 4.0 - NSQLChangeMonitor and NOraChangeMonitor are Enterprise Edition features and are unintentionally exposed in NCache Professional Edition.

## Release Notes NCache 3.8 (Service Pack 3)

Tuesday, Nov 02, 2010

### Introduction

In this release of NCache 3.8 Service Pack 3 (SP3), NCache Enterprise Edition is separated into two products named as 'NCache for .NET' and 'NCache for Java'. Previously 'NCache' had both components combined into a single package.

Two new features are also added in 'NCache Java Client'.

### Enhancements and New additions

Following features are added in 'NCache Java Client' in this release.

#### 1. Database Dependency

Now you can add items in cache with 'database dependency' from your java application. Any change in related database invalidates the cache entry, and your application will receive the notifications about this change.

#### 2. Streaming API

NCache Java Client has added streaming support in the API where you can read and write binary data stream in the cache.

### List of Bugs Fixed:

[738301](#) FIX: On adding new node in cluster, new node info was not added in "client.nconf" file on all cluster nodes.

---

#### APPLIES TO:

- NCache Enterprise Edition 3.8.x

## Release Notes NCache 3.8 (Service Pack 2)

Monday, Oct 10, 2010

### Introduction

NCache 3.8 Service Pack 2 (SP2) contains important fixes and enhancements that have been reported by customers. It is a recommended upgrade for all users of NCache 3.8.

### Enhancements and New additions

Following features are added in this release.

#### 3. License Support for Xen VM

NCache now support licensing on Xen VM environments.

#### 4. Visual Studio 2010 Assembly Reference solution

Visual Studio 2010 shows a warning message when you add reference to NCache assemblies and then the build also fails. This only happens if the application target frame work is .NET 3.5. This is a known issue of Visual Studio that it gets confused when you have the same assemblies available in the GAC 2.0 and GAC 4.0 and for dependent assemblies it tries to load them from GAC 4.0 which results in the warning message.

We have resolved this issue by copying all the assemblies in the NCache bin/assembly folder and now Visual Studio does not have to locate the dependent assemblies in GAC. This has resolved the issue.

### List of Bugs Fixed:

[738211](#) FIX: ViewState Caching does not work with AJAX controls especially GridViewPager.

[738210](#) FIX: Session Store Provider throws an exception "The SessionStateStoreData returned by ISessionStateStore has a null value for items" if the session is empty.

[738209](#) FIX: ViewState assemblies were build with wrong version in NCache Enterprise Developers installation.

[738208](#) FIX: There was a performance issue in the feature object query. The search result was slower when you have a large set of items in the cache.

[738207](#) FIX: NHibernate sample has a build issue in NCache Professional Edition.

[738206](#) FIX: NCache Manager throws an object reference not set error if you enable the security without specifying users.

[738205](#) FIX: NCache Manager throws an object reference not set error if you change the bind to IP (using the option "Select Network Interface Card") for local cache server. This option is now removed from local cache server.

[738204](#) FIX: NCache Manager fails to load ReadThru/WriteThru Provider if the provider is an exe file. When you deploy the ReadThru/WriteThru provider using the "Deploy Provider" button, NCache Manager rename the file extension to .dll

[738203](#) FIX: NCache Professional does not allow remote client connections. This happens if the remote client is installed with NCache Professional Remote Client installation.

[738202](#) FIX: NCache integration with Microsoft Enterprise Library v4.1 returns Boolean value for indexer method `Cache[ "Key" ]` instead of the actual object.

[738201](#) FIX: Multiple bugs are fixed in JavaScript and CSS Minification. The JavaScript Minification fails if the rendered output contains .axd files in the JavaScript tag, and if the `<script>` tag ends with `</Script>`. Similarly, the CSS minification fails if relative paths are used for loading images. Now the relative paths in CSS are converted to absolute path.

---

**APPLIES TO:**

- NCache Enterprise Edition 3.8.x
- NCache Professional Edition 3.8.x

## Release Notes NCache 3.8 (Service Pack 1)

Tuesday, August 24, 2010

### Introduction

NCache 3.8 Service Pack 1 (SP1) contains important fixes and enhancements. The most important and demanded feature added in this release is the support of .Net frame work 4.0

The API is completely compatible with the 3.8 release version and applications can upgrade without re-building/re-compiling the application.

### Enhancements and New additions

Following features are added to this service pack 1.

**1. .Net 4.0 support available.**

The code base of NCache cache server has been converted to .NET 4.0 and the NCache client is available in both .NET 2.0 and 4.0 versions.

**2. Improvement in client cache management**

There is an improvement in client cache management through NCache Manager where project files will contact to client nodes on refresh option and this has improved fast loading of Ncache Manger project file.

**3. Samples are now builded with Visual studio 2008.**

Ncache samples are now builded with visual studio 2008.

**4. VeriSign issue.**

VeriSign issue, default in service configuration file should be generatePublisherEvidence enabled = "false"/>. The enhancement is made.

**5. ReadThru Interface improved**

ReadThru provider interface signature has been modified to support maximum features of NCache. There is new structure introduced under the namespace "Alachisoft.NCache.Runtime.Caching" called `ProviderCacheItem` which is similarly to the `CacheItem`. You can now easily specify expirations, tags, eviction hints, dependencies etc.

New interfaces

```
public void LoadFromSource(string key, out ProviderCacheItem cacheItem)
public Dictionary<string, ProviderCacheItem> LoadFromSource(string[] keys)
```

**6. CacheLoader Supports IsResyncExpiredItem**

Now, you can specify `IsResyncExpiredItem` property in Cache Loader so that the expired items can be reloaded automatically.

**7. NHibernate Integration**

NCache is not supporting the latest version of NHibernate 2.1.2. We have also added region support in this release. NHibernate sample application is also modified with NHibernate regions support.

### List of Bugs Fixed:

**738108** FIX: AutoStart fails to start the cache if the cache is under heavy load. Now the AutoStart starts the cache in Asynchronous mode.

**738107** FIX ViewState assemblies were built with wrong version in NCache Enterprise Developers installation.

**738106** FIX: CAB Integration does not work in NCache professional Developers installation because of aggregate dependency.

**738105** FIX: Create Cache tool bug fix.

**738104** FIX: NHibernate integration requires complete DateTime format for absolute expiration which is wrong in. Now, you can specify absolute expiration in terms of seconds.

**738105** FIX: There was a serialization bug in Session Store Provider which occurs in rare scenarios only.

**738103** FIX: SQL Dependency bug fix in NHibernate where client was unable to use queries with composite key.

**738102** FIX: Client cache does not work with NHibernate Integration.

**738101** FIX: GetByTag performance is slow for large number of items.

**736230** FIX: Object query returns empty result if the item was automatically reloaded (IsResyncExpiredItem is true) through the ReadThru provider.

---

**APPLIES TO:**

- NCache Enterprise Edition 3.8
- NCache Professional Edition 3.8

## Release Notes NCache 3.8

Monday, June 7, 2010

### Introduction

NCache 3.8 contains new important additions and enhancements based on the customer's feedback. The API backward compatibility is the most important concern for most of the customers and keeping this in mind we have added a new protocol for API compatibility. From now (3.8) onwards all the new releases will be automatically compatible with older versions of NCache (starting from 3.8).

### Enhancements and New additions

Following features are added in this release.

#### 1. LINQ Support in NCache

NCache provides LINQ integration with the help of **IQueryable** interface which allows the cached items to be searchable. NCache support both Lambda Expressions and LINQ operators for querying cached items.

#### 2. Entity Framework (EF) Caching

NCache provides seamless integration with EF caching where it gets plugged-in at the ADO level and lets your application use distributed caching without any code change. You only need make changes in the application configuration file.

#### 3. .NET Cache Provider 4.0

NCache provides integration with .NET cache provider 4.0. NCache also provides different **Change Monitors (file based, key based, database dependency)** for managing cache dependencies.

#### 4. Backward Compatibility Client API/Support

NCache now follows backward compatibility protocol and in future version application will be able to connect to newer versions without upgrading the clients.

#### 5. Streaming API

NCache has added streaming support in the API where you can read and write binary data stream in the cache.

#### 6. Java/CSS minification

NCache combines multiple JavaScript files and CSS files into a single resource file and store it in the cache. It also replaces the rendered output with single HTTP reference for all CSS and JS files so that browser can make a single call for loading all the resources. This helps improve your application response time.

#### 7. ViewState Caching

NCache replaces the long ViewState string into a smaller one and sends it to the client. This helps improve the application performance and save bandwidth.

#### 8. New NCache Monitoring Tool

#### 9. NCache Email Alert System

You can now receive alert through emails on certain cache events like "State Transfer", Cache Stopped, Member Left, Member Joined etc.,

#### 10. Cache Meta information API

This API allows Meta information about cache items like LastAccessed Time and Creation DateTime.

**11. Multi-ReadThru/Write Thru support**

NCache allows multiple readthru and writethru providers. NCache Manager automatically deploys the data source assemblies into the deployed folder so you don't have to manually copy the provider assemblies in NCache service folder.

**12. Partitioned Replica Synchronous replication**

Partitioned Replica Topology now supports Synchronous replication

**13. Security Configuration command line tool**

You can now configure security through command line tool

**14. Cache Config Upgrade tool**

If you have an old cache config.nconf file then you can upgrade it by using this tool.

**15. Locking support with GetCacheItem**

**16. Remote Client Management from NCache Manager**

Now you can configure Remote Clients from NCache Manager and can easily change individual client settings all from a single point.

**17. Client-Cache Management from NCache Manager**

Client Cache can also be managed from NCache Manager.

**18. Client/Client Cache management command line tools**

You can also add remote clients and client-cache through command line tools.

**19. New interface for ReadThru/WriteThru providers.**

Now you can specify your cache provider assemblies through a new wizard where you can pick your assembly. Now, you don't have to type assembly information manually.

**20. Automatic deployment of ReadThru/WriteThru providers**

NCache Manager provides the automatic deployment of data source providers.

**21. Dynamic Compact Serialization Support (no code change required)**

Now you don't need to implement any ISerializable or IDeserializable Interfaces for compact serialization. You only need to register the compact types in NCache Manager and NCache automatically serialize the types at runtime.

**22. Security Enhancement (Security Management from NCache Manager)**

Security can be configured from NCache Manager now.

## Release Notes NCache 3.6.2 (Service Pack 3)

Thursday, April 21, 2010

### Introduction

NCache 3.6.2 Service Pack 3 (SP3) contains important fixes and enhancements that have been reported by customers. The most important and demanded feature added in this release is the support of Java session caching for J2EE platforms. Now, you can easily use NCache as a session store for your Java applications without making any code change. You only need to add a Java session filter in your web.xml file.

Those using NCache 3.6.x version can upgrade without re-building/re-compiling the application.

### Enhancements and New additions

Following features are added to this service pack 3.

**1. Java Session Support for WebLogic (no-code change required)**

NCache Java Session Provider is supported for WebLogic and can be used on any platform of J2EE (web servers), if underlying platform (web server) follows the Java Servlet 2.3 (or later) specification. Java Sessions are supported on both Linux and Windows platforms.

**2. Java Session Support for JBoss (no-code change required)**

NCache Java Session Provider is supported for JBoss on both Linux and Windows platforms.

**3. Java Session Support for WebSphere (no-code change required)**

NCache Java Session Provider is supported for WebSphere on both Linux and Windows platforms.

**4. Java Session Support for ApacheTomcat (no-code change required)**

NCache Java Session Provider is supported for Apache-Tomcat on both Linux and Windows platforms.

**5. NCache support for Server GC**

There are two flavors of Garbage Collector based on server operating system and workstation called "Server GC" and "Workstation" respectively. Unless it is specified .NET framework uses workstation GC by default even it is running under server operating system. Now, NCache allows you to specify GC mode depending on your operating system.

Server GC is designed for maximum throughput, and scales with very high performance. NCache is now by default configured to use Server GC instead of Workstation GC. You can change this mode from Alachisoft.NCache.Service.exe.config

```
<add key="NCacheServer.EnableForcedGC" value="true" />
<add key="NCacheServer.ForcedGCThreshold" value="80" />
```

### List of Bugs Fixed:

[736228](#) FIX: NCache installation fails on Windows 7 because of new security model in Windows 7. This has been fixed now.

[736227](#) FIX: There was a bug in Client Queue Counter which becomes negative at some particular stage. This bug has been fixed.

[736226](#) FIX: There was a bug in item expiration in Java API which has been fixed now.

---

**APPLIES TO:**

- NCache Enterprise Edition 3.6.x

## Release Notes NCache 3.6.2 (Service Pack 2)

Monday, Jan 11, 2010

### Introduction

NCache 3.6.2 Service Pack 2 (SP2) contains important fixes and enhancements that have been reported by customers. It is a recommended upgrade for all users of NCache. Those using NCache 3.6.x version can upgrade without re-building/re-compiling the application.

### Enhancements and New additions

Following features are added to this service pack.

#### 1. Logging of Important events into event viewer

An event will be logged in windows event viewer and cache-error log when

- o you stop or start a cache
- o cache fails to start
- o cache node joins/leaves a cluster
- o state transfer start or stop.
- o client connect or disconnect a cache
- o cache memory crosses a certain threshold specified in service config file.
- o you are on last 10 days of NCache evaluation and a log warning event will be logged per day until you extend the evaluation or activate NCache with a license key.

These events are categorized as information, warning and information.

```
<add key="NCacheServer.EventLogLevel" value="error | warning| all"/>
```

User can specify only one of the following levels:

**error:** Only the error events will be logged

**warning:** Both error and warning events will be logged.

**all:** This level allows events of all categories to be logged. This is the default level.

Client Connection Log entry for event viewer/log-file can be enabled from NCache Service configuration file. By default this option is disabled but you can enable it by modifying the following attribute in NCache service config file "Alachisoft.NCache.Service.exe.config".

```
<add key="NCacheServer.LogClientEvents" value="false" />
```

**NOTE:** The above change requires NCache service to restart.

#### 2. Oracle 11g is supported

NCache now supports Oracle 11g for database dependency and synchronization feature.

**List of Bugs Fixed:**

**736225** FIX: Some of NCache clients stop responding when the cache cluster is under stress and you add a new caching server or the cache servers are using shared network card on VM.

**736224** FIX: NCache service (Alachisoft.NCache.Service.exe) and NCache Manager stop responding when the cache is under stress and you add a new caching server or the cache servers are using shared network card on VM.

**736223** FIX: NCache service stops responding due to Hashmap corruption. This is similar to the above mentioned service hanging scenario, however in this scenario, service hangs for indefinite time.

**736222** FIX: Cache stops expiring items from the cache if a WMI call does not respond for a long period of time because of WMI unresponsive state.

**736221** FIX: Client unbalancing in Partitioned and Partitioned Replica on node joining is fixed now. This only happens in very rare situations.

**736220** FIX: Clients were not able to get real performance benefits of client-optimization feature of Partitioned Replica if the clients are 32-bit and cache is 64bit or vice versa.

**736219** FIX: Cache state transfer (starts when a cache node joins the cluster) does not transfer those items which have multi-level key dependent items.

**736218** FIX: Java client environmental issues have been fixed. Now you can pass client.nconf file path from API. Java client now supports both Windows and Linux environment.

**736217** FIX: Java sample application bugs are fixed now

**736216** FIX: NCache service does not start if the socket call fails to bind itself network card. NCache now tries to establish the connection without binding.

**736215** FIX: When you unplug the network cable Client hanging issue when cable is unplugged during stress is fixed now. This normally happened in replicated topology and the client operations were hanged for sometime.

**736214** FIX: NCache tag query throws "Null Reference" in partitioned and partitioned replica topologies when used with LIKE operator and the tag does not exist.

**736213** FIX: NCache throws "Data group mismatch" exception when you update and item by calling INSERT method with locking and group parameters.

---

**APPLIES TO:**

- NCache Enterprise Edition 3.6.x
- NCache Professional Edition 3.6.x

## Release Notes NCache 3.6.2 (Service Pack 1)

Tuesday, Oct 06, 2009

### Introduction

NCache 3.6.2 Service Pack 1 (SP1) contains important fixes and enhancements that have been reported by customers. It is a recommended upgrade for all users of NCache. Those using NCache 3.6.x version can upgrade without re-building/re-compiling the application.

### Enhancements and New additions

Following features are added to this service pack.

#### 1. Cluster Rejoining feature

To overcome socket breaking issues within cluster nodes, NCache has a new connection retry logic which is configurable in config.nconf in cluster configuration.

```
connection-retries="10" connection-retry-interval="1secs"
```

In case a socket breaks due to Network problems, NCache will try to re-establish the connection after every connection-retry-interval as many times as connection-retries. [This requires a restart of NCache service after update.](#)

#### 2. Addition of NCache Client side counters

NCache client side counters are introduced to detect and debug client side issues. These counters appear in Windows perfmon counters in category 'NCache Client'.

#### 3. Asynchronous startup of caches in Autostart

NCache autostart feature is enhanced and starting up of various user specified caches with NCache service start up, is made asynchronous. If some caches are corrupt and are not started, a warning is logged but NCache service starts up normally.

#### 4. Client-cache-sync mode description

Client cache sync mode description is added in NCache Help.

#### 5. Memory estimation and warning mechanism in cluster configuration

Description about setting up cluster memory, its usage and estimation is added in cluster configurations.

#### 6. NCache Installation in Admin mode

NCache installation wizard warns the user who is not an admin. Administrative privileges and permissions are required for the user to install NCache on a system.

### List of Bugs Fixed:

© Copyright 2005-2012 by Alachisoft. All rights reserved

**736212** FIX: Items with same Tags if added and removed frequently resulted in corrupting the tag. Items exist in the cache but not gettable with their tag.

**736210** FIX: The memory leakage related with LOH results in high memory consumption in case of large objects (more than 80KB) which are not garbage collected.

**736209** FIX: NCache generates extra log entries in NCache log files which result in huge log files. These extra logs are removed.

---

**APPLIES TO:**

- NCache Enterprise Edition 3.6.x
- NCache Professional Edition 3.6.x

## Release Notes NCache 3.6.2

Thursday, July 09, 2009

### Introduction

NCache 3.6.2 contains a number of useful enhancements and bug fixes that have been reported by customers. It is a recommended upgrade for all users of NCache. Those using NCache 3.6.x version can upgrade without re-building/re-compiling the application.

### Enhancements and New additions

Following features are added to this version.

#### 1. Log Traces

Log traces are added for the following events

1. When a node successfully joins a node in the cluster.
2. When a node can not join a node in the cluster.
3. When a node leaves the cluster.

#### 2. Polling based database dependency improvements

Polling based database dependency has the following improvements.

4. User can specify the 'db-cache-key' other than the 'cache-key' to add to the ncache\_db\_sync table.
5. Resync-expired-items option is now available for this dependency.

#### 3. Client.nconf can be opened in notepad

#### 4. Improved Error messages

Error messages are improved when service fails to start because of unavailable bind-to-ip addresses.

### List of Bugs Fixed:

[736208](#) FIX: Expiration problem that arises in around 20-30 days of using NCache results in items occupying memory leading to high memory usage.

[736207](#) FIX: Client.nconf is rewritten by NCache service restart and old changes are lost.

[736206](#) FIX: Groups and tags information is not preserved if the items are read-thru from a datasource. Items are loaded in the cache but do not have the tag or group already specified.

[736205](#) FIX: IIS worker process high CPU consumption bug is fixed

---

#### APPLIES TO:

- NCache Enterprise Edition 3.6.x
- NCache Professional Edition 3.6.x

## Release Notes NCache 3.6.1

Thursday, March 26, 2009

### Introduction

NCache 3.6.1 contains a number of useful enhancements and bug fixes that have been reported by customers. It is a recommended upgrade for all users of NCache. Those using NCache 3.6.x version can upgrade without re-building/re-compiling the application.

### Enhancements and New additions

Following features are added to this version.

#### 1. NCache client rebalancing among cache server nodes

Now the NCache client connections are automatically rebalanced among the cluster nodes when a new node joins the cache. So now you do not have to worry any more about client load balancing on cache servers.

#### 2. New Counters for performance monitoring

New counters for performance monitoring are added in the 'NCache' category in Windows Perfmon counters.

#### 3. Improvements in DB dependency

NCache now supports the use of Stored-Procedures and Command Object in DB Dependency.

### List of Bugs Fixed:

[736204](#) FIX: NCache service constantly consumes CPU if there is one item left in the cache with the SQL Yukon dependency and the dependency is triggered.

[736203](#) FIX: AVG counters show incorrect values for Average Add/sec, Update/sec and Remove/sec.

[736202](#) FIX: Memory leakage and performance issue related with Least Recently Used and Least Frequently Used eviction policies (LRU/LFU).

[736201](#) FIX: Bind to IP Dialog corrupts the display if there are more than 2 network cards on a system.

[736200](#) FIX: NCache Service consumes CPU at some fixed interval even when there is no activity on the cache after a load test is stopped after running for hours.

---

### APPLIES TO:

- NCache Enterprise Edition 3.6.x
- NCache Professional Edition 3.6.x