

Software AG

**webMethods Nirvana Messaging 7.0
Release Notes**

webMethods Nirvana Messaging

*The market-leading universal messaging
platform for real-time streaming across
Enterprise, Web and Mobile.*

Software AG
Tel: +44 (0) 20 7375 7400
Email: nirvana-support@softwareag.com

Table of Contents

<i>Nirvana 7.0 SP 16, Build 7.0.16.0.83059</i>	4
Bug Fixes and Enhancements.....	4
<i>Nirvana 7.0 SP 15, Build 7.0.15.0.81322</i>	5
Bug Fixes and Enhancements.....	5
<i>Nirvana 7.0 SP 14, Build 7.0.14.0.79224</i>	6
Bug Fixes and Enhancements.....	6
<i>Nirvana 7.0 SP 13, Build 7.0.13.0.77988</i>	7
Bug Fixes and Enhancements.....	7
<i>Nirvana 7.0 SP 12, Build 7.0.12.0.77653</i>	8
Bug Fixes and Enhancements.....	8
<i>Nirvana 7.0 SP 11, Build 7.0.11.0.76909</i>	9
Bug Fixes and Enhancements.....	9
<i>Nirvana 7.0 SP 10 Build 7.0.10.0.76208</i>	10
Bug Fixes and Enhancements.....	10
<i>Nirvana 7.0 SP 9 Build 7.0.9.0.13392</i>	11
Bug Fixes and Enhancements.....	11
<i>Nirvana 7.0 SP 8, Build 7.0.8.0.13258</i>	12
Bug Fixes and Enhancements.....	12
<i>Nirvana 7.0 SP 7, Build 7.0.7.0.13167</i>	13
Bug Fixes and Enhancements.....	13
<i>Nirvana 7.0 SP 6, Fix 1, Build 7.0.6.1.13061</i>	14
Bug Fixes and Enhancements.....	14
<i>Nirvana 7.0 SP 6, Build 7.0.6.0.13031</i>	15
Bug Fixes and Enhancements.....	15
<i>Nirvana 7.0 SP 5, Build 7.0.5.0.12837</i>	16
Bug Fixes and Enhancements.....	16
<i>Nirvana 7.0 SP 4, Nirvana 7.0.4.0.12638</i>	17
Bug Fixes and Enhancements.....	17
<i>Nirvana 7.0 SP 3, Nirvana 7.0.3.0.12415</i>	19
Bug Fixes and Enhancements.....	19
<i>Nirvana 7.0 SP 2, Fix 2, Build 7.0.2.2.12119</i>	21
Bug Fixes and Enhancements.....	21
<i>Nirvana 7.0 SP 2, Fix 1, Build 7.0.2.1.12094</i>	23
Bug Fixes and Enhancements.....	23
<i>Nirvana 7.0 SP 2, Build 7.0.2.11880</i>	24
Bug Fixes And Enhancements	24
<i>Nirvana 7.0 SP 1, Build 7.0.1.0.11750</i>	25
Bug Fixes And Enhancements	25
<i>Nirvana 7.0.0, Build 7.0.0.0.11318</i>	26
Bug Fixes And Enhancements	26
<i>Nirvana 7.0 Feature Summary</i>	30
<i>Nirvana 7.0 EA2 (11091) to Nirvana 7.0 Release</i>	30
Enhancements	30
Nirvana Server	30
Enterprise Manager.....	31

Javascript API	31
<i>Nirvana 7.0 EA1 (10869) to Nirvana 7.0 EA2</i>	32
Shared Memory	33
Data Group Fanout.....	33
New JavaScript API.....	33
Enterprise Manager	33
Nirvana 7.0 Early Access 1 (EA1) – What’s New.....	34
Multicast Support.....	34
MQTT Support.....	34
HTML 5 JavaScript.....	35
Performance Improvements	35
Data Routing Between Nirvana Clusters	35
Nirvana Admin API Enhancements	35
Nirvana Client API Enhancements	35
Legal notices.....	36

Nirvana 7.0 SP 16, Build 7.0.16.0.83059

Bug Fixes and Enhancements

Server

- i. NUM-3709 (Defect) – Ensure data group ID's are recovered in cluster recovery, also ensure we keep the latest connection cluster wide for data streams otherwise we could have a stale state

JavaScript

- i. NUM-5146 (Defect) – When recreating an XHR_STREAMING_CORS driver make sure we schedule the reconnect task just once

Java

- i. NUM-5307 (Defect) – Ensure we tidy up when adding async consumers if we have an issue when requesting the subscribe from the server

Nirvana 7.0 SP 15, Build 7.0.15.0.81322

Bug Fixes and Enhancements

Server

- i. NUM-4597 (Defect) - Fixes a problem with HTTP header preparation where pub hostname might contain invalid JSON characters.
- ii. NUM-4865 (Defect) - When a HTTP header was split over multiple TCP packets a misaligned marker would append two headers into one. This could potentially then disallow clients to operate normally as there is various configurations pulled out of the headers.
- iii. NUM-4758 (Defect) - Ensure client exceptions generated by the server (e.g. security exceptions) are propagated to JavaScript clients during server handshake.

Nirvana 7.0 SP 14, Build 7.0.14.0.79224

Bug Fixes and Enhancements

Server

- iv. NUM-3798 (Defect) - Resolved an issue where the send and receive buffer sizes for server connections would not scale beyond 64KB.
- v. NUM-3821 (Defect) - Resolved a case where an exception would be raised before completing a store removal during cluster restart.
- vi. NUM-3921 (Defect) - Require a higher logging level to produce the "Protocols Enabled" logging message.
- vii. NUM-4005 (Defect) - Resolve issue where float data with digits with a larger size than 7 digits would not be correctly sent to a JavaScript client.
- viii. NUM-3962 (Defect) - Resolve a timing issue related to changing the structure of Registered Events on DataGroups causing messages to not be received.

Java

- i. NUM-3798 (Defect) - Resolved an issue on the client where the send and receive buffer sizes would not scale beyond 64KB.

Tools

- i. NUM-4218 (Defect) - The "nexportrealm" sample application reports that it cannot locate the application main class and fails to start. The issue is caused by obfuscation of UM API classes and is now resolved.

Nirvana 7.0 SP 13, Build 7.0.13.0.77988

Bug Fixes and Enhancements

Server

- ix. NUM- 3628 (Defect) - Resolved a regression introduced in Nirvana 7.0 SP 10 where stream connection callbacks were not delivered in certain circumstances to clients.
- x. NUM- 3164 (Defect) - Resolved an issue where Data group permissions would not be persisted in certain circumstances, therefore after a restart these permissions would be lost.

Java

- ii. NUM- 3673 (Defect) - Resolved an issue where initializing an nRealmNode with one realm in a cluster down would cause the AdminAPI to leak a listener after each attempt to connect to the realm currently down.

Nirvana 7.0 SP 12, Build 7.0.12.0.77653

Bug Fixes and Enhancements

Server

- i. NUM- 3544 (Defect) – Frequent GCs observed in cluster, due to leak in websocket connections.
- ii. NUM- 3621 (Defect) – Memory leak in UM cluster.

Enterprise Manager

- i. NUM- 3612 (Defect) - It is not possible to modify cluster members through the Enterprise Manager, unless the user has explicitly added a connection to that specific server node in the Enterprise Manager.

Nirvana 7.0 SP 11, Build 7.0.11.0.76909

Bug Fixes and Enhancements

Server

- ii. NUM- 3162 (Defect) – Datastreams are now properly maintained on all cluster nodes when a node gets disconnected from the cluster.
- iii. NUM- 3342 (Defect) - When a HTTP socket receives a native protocol connect there was one situation where we were doing HTTP 1.1 communication we incorrectly did not downgrade the socket to HTTP 1.0 as required. This is now resolved. Also we now allow the header Keep-Alive to be disabled with a system property

JavaScript

- i. NUM- 3433 (Defect) - Version and build information was not set correctly in the nirvana.js library. The issue is resolved. The JavaScript library is properly stamped with the version, build number and build date.

Nirvana 7.0 SP 10 Build 7.0.10.0.76208

Bug Fixes and Enhancements

Server

- i. NUM-3099 (Code Roll-In) – JMX- Fixed updating the channel status. The status was not updated properly- for 'CurrentConnections' and 'NoOfEvents' attributes. This is now resolved.
- ii. NUM-3135 (Code Roll-In) - A connection to the realm server could be put indefinitely in recovery mode when a high water mark of received messages has been reached and would stop receiving messages. This behavior is now fixed and the connections are returned to normal state as expected. This is now resolved.

Tradespace Demo

- i. NUM-2107 (IOS client)- Added a "ping" server method to the nSession class (ie. nSession::pingServer), which can be used to accurately determine the state of a socket (essentially, we write to AND read from a socket in one call).
Customer can use the result of this "ping" to aid them in some aspect of their application design (ie. exception handling).
Further documentation on the method can be found in the nSession header file.
- ii. NUM-2804 (Android client) – The Tradespace Android Mobile App failed to display any data. The realm URL was incorrect. We fixed it to point to the right realm.

Nirvana 7.0 SP 9 Build 7.0.9.0.13392

Bug Fixes and Enhancements

Server

- i. NUM-2447 (Code Roll-In) – Issue with event replay to named object subscribers after full cluster restart. The issue reported demonstrated that after a full cluster restart, events could be replayed to durable consumers. This is now resolved.
- ii. NUM-2473 (Code Roll-In) - Issue with Data Group consumers receiving duplicate events in the case that the Publisher is slow. This would only happen in conflated Data Groups where the conflation period was smaller than the time between published events.
- iii. NUM-2460 (Code Roll-In) - Integrated functionality for multi-fragment Web Socket events. Previously only the first fragment of the message would be correctly processed.
- iv. NUM-2495 - Resolved an issue where the named subscriber storage file could become corrupted when subscribers are deleted and recreated.
- v. NUM-2723 - Resolved an issue where expired events that are rolled back are resent to subscribers. The TTL for these messages was reset; it is now preserved through the rollback process.

Enterprise Manager

- i. NUM-2653 - Disk usage displayed in the Monitoring -> Top -> Top Channel Usage panel were labeled as kilobytes but were displayed in bytes. The panel now displays values as kilobytes.

C#

- i. NUM-2312 - Resolved issue where multiplex connections could not be established when using SSL connections. This was resolved by creating separate reference objects for sessions to lock when reading or writing to the underlying driver.

Tradespace Demo

- i. NUM-2742 - Resolved an issue where Tradespace would attempt to connect to an unreachable host to download news information.

Nirvana 7.0 SP 8, Build 7.0.8.0.13258

Bug Fixes and Enhancements

Server

- i. NUM-2205 (Code Roll-In) – Memory not being released by server. Issue was regarding the recovery of data groups during cluster reformation. The flow control object sometimes would not be created on recreation of the recovered data group state, leading to events failing to be cleared up for specific connections. The resolution for this was to ensure correct creation of the flow control object, and setting a maximum event queue size for connections before termination of the connection.

Enterprise Manager

- i. NUM-2049 – When creating / deleting a cluster through the Admin API, any connected Enterprise Managers did not reflect the correct state.
Resolved

Nirvana 7.0 SP 7, Build 7.0.7.0.13167

Bug Fixes and Enhancements

Server

- i. NUM-1474 – Issue with occasional pauses to delivery of messages. Fix was related to setting of ACLs on slave nodes while consumers were actively connected which has now been optimised.
- ii. NUM-2108 (Code Roll-In) – When a node is added to a cluster, only events published after the addition are republished to the slave. Issue resolved.
- iii. NUM-1545 – Auto generated thread dumps in the cluster from close pool. Resolved contention issue during closure of client connections.
- iv. NUM-1020 – Nirvana clustering issue, relates to NUM-2108, resolved.
- v. NUM-2050 – 3 Logout messages in log file for single client connection. This issue has been resolved by reducing the possible paths through to logout calls.

C# Client API

- i. NUM-1779 – `ArgumentOutOfRangeException` thrown in C# client during removal of stream from Data Group. Resolved issue with additional synchronisation around the `SortedList` object containing streams during the data stream removal event being processed by the client API.

Nirvana 7.0 SP 6, Fix 1, Build 7.0.6.1.13061

Bug Fixes and Enhancements

Server

- i. Poodle Fix for SSL. Disable SSL by default, only use TLS.

Nirvana 7.0 SP 6, Build 7.0.6.0.13031

Bug Fixes and Enhancements

Server

- i. The log file rolled over twice when log rolling was scheduled using scheduler script in the script tab. The first roll in happens one minute before the actual scheduled time and the second roll in happens at the actual scheduled time.

Java

- i. Fixed javax.naming implementations for the JNDI API.
- ii. Fix when subscribing using selectors at very high rates.
- iii. Fix duplicate callbacks on nDataGroupListeners when calling createDataGroup.
- iv. Enable support for JDK v1.8

Nirvana 7.0 SP 5, Build 7.0.5.0.12837

Bug Fixes and Enhancements

Server

- i. Fix issue where the log file would roll twice on realm startup.
- ii. Fix issue with multiple logfile rolls when using the scheduler scripts.
- iii. Resolve issue where the GC() function would be manually called too often.
- iv. Fixed server fault where nDataGroupListeners would not receive a deletedStream callback from different nodes on a cluster.
- v. Fixed a casting exception by checking if we are using a MappedStream
- vi. Fixed named object priority subscription with filters by changing the base connection type to use a filtered connection.
- vii. Fix concurrency issue within the merge engine during snapshot writes
- viii. Enable support for JDK v1.8
- ix. Fix clusterwide issue with revocation of realmwide ACLs impacting performance when causing queue subscribers to disconnect.

Java

- i. Fixed javax.naming implementations for the JNDI API.
- ii. Fix when subscribing using selectors at very high rates.
- iii. Fix duplicate callbacks on nDataGroupListeners when calling createDataGroup.
- iv. Enable support for JDK v1.8

C++

- i. No longer statically link to openssl within the iOS libraries.
- ii. Fix duplicate callbacks on nDataGroupListeners when calling createDataGroup

C#

- i. Fix duplicate callbacks on nDataGroupListeners when calling createDataGroup
- ii. Fix incorrect number of callbacks on delete notifications for nDataGroup listeners

Enterprise Manager

- i. Fix issue with duplicate instances of DataGroups and Channels being rendered by the Enterprise Manager.

JavaScript

- i. Resolved an issue which would occur when a load balancer was reusing http connections to the UM server from different domains externally. This was causing problems with CORS origin checks.

Nirvana 7.0 SP 4, Nirvana 7.0.4.0.12638

Bug Fixes and Enhancements

Server

- i. Recover ACLs when synching the group
- ii. Remove groups which are not on the master recovery list
- iii. Fix recovery of sparsely distributed events
- iv. Fix issue with conflation deltas being marked as snapshots
- v. Fix logfile rollover issue
- vi. Ensure that snapshots always replace existing events in the store
- vii. Set config as update when initialising because we are master
- viii. Fix blocking comms issues between slaves
- ix. Only the master should set the UID

Java

- i. Allow session to add/remove listeners once default DG has been initialised

C#

- i. Update caching structure to align with Java
- ii. Change Silverlight helper classes to use global namespace
- iii. Fix datagroup listener issues
- iv. Fix bulk datagroup creation, with conflation attributes and multicast

C++

- i. Don't send duplicate deltas to server

JavaScript

- i. Fix issue with XHR_STREAMING_CORS and XDR_STREAMING going through rapid reconnect/disconnect states

Nirvana 7.0 SP 3, Nirvana 7.0.3.0.12415

Bug Fixes and Enhancements

Server

- i. Considerable performance improvement of stream removal when DataGroups are placed in hierarchy.
- ii. Configuration changes to allow for selection of different cluster pipeline mechanisms across servers. The default mechanism has been changed to be the ThreadedPipelineProcessor.
- iii. Ensure that groups that have a parent do not incorrectly trigger stream removal on those parent groups.
- iv. Fix for threading issue on cluster state change that required realms to be restarted to re-join the cluster.
- v. Change to ensure that priority listeners always receive messages across a cluster.

Java

- i. Improvements to requests to add listeners to groups.
- ii. Fix for clients holding onto objects that were purged from a Channel.
- iii. Ensure that a group cannot be added to another group as a stream. Now the user will receive an exception message if they attempt to do this.
- iv. Correct API documentation to correctly reflect the behaviour of certain DataGroup methods.
- v. Fix race condition with DataGroup objects when constantly deleting and recreating the same group object on a realm.
- vi. Fix race condition with DataGroup objects when constantly deleting and recreating the same group object on a realm.

C#

- i. Fixed issue where occasionally a call to remove a DataGroup would remove a different group when specific combinations of groups were involved within the C# API.
- ii. Improvements to requests to add listeners to groups.
- iii. Improvements to the speed of cached group lookups for DataGroups.
- iv. Ensure that a group cannot be added to another group as a stream. Now the user will receive an exception message if they attempt to do this.
- v. Correct API documentation to correctly reflect the behaviour of certain DataGroup methods.
- vi. Fix race condition with DataGroup objects when constantly deleting and recreating the same group object on a realm.

C++

- i. Improvements to requests to add listeners to groups.
- ii. Change printing mechanism in connection object.
- iii. Ensure that a group cannot be added to another group as a stream. Now the user will receive an exception message if they attempt to do this.
- iv. Correct API documentation to correctly reflect the behaviour of certain DataGroup methods.
- v. Fix race condition with DataGroup objects when constantly deleting and recreating the same group object on a realm.

Enterprise Manager

- i. Fixed issue where joins between queues were not being imported to realms.

IPhone

- i. SSL Certificates on iPhone devices are now treated consistently with other APIs.

JavaScript

- i. Add getDataStreamID() method to session object. DataStream IDs should remain the same after any unexpected disconnection/reconnection.
- ii. If a session configuration object contains a property named "applyUserAgentHeuristics" with the boolean value true, then additional heuristics will be applied to disable drivers that are known not to work in specific browsers.
- iii. Corrected documentation for the Event Dictionary's getArrayType() method.

Nirvana 7.0 SP 2, Fix 2, Build 7.0.2.2.12119

Bug Fixes and Enhancements

Server

- i. Fix pauses in heartbeat distribution by ensuring relevant thread hands off tasks to correct thread pools.
- ii. Fixed issue where lack of notification could stall threads handling JavaScript clients.
- iii. Ensure that if registered event data is processed whilst the channel is in recovery, merging happens in the correct order.
- iv. Add auditing for data group operations

Java

- i. Fix erroneous caching of data streams when there is no listener registered.
- vii. Maintain the ACLs when copying a queue

C#

- i. When using registered events, ensure that only key/value pairs that have changed are sent to the server.
- ii. Improve support for 3rd party cookies

Flex

- i. When using registered events, ensure that only key/value pairs that have changed are sent to the server.
- ii. Improve behaviour when receiving large events

Enterprise Manager

- i. Allow cluster deletion when migrating cluster resources to local resources.
- ii. Improve cluster deletion dialog.

JavaScript

- i. Improvements to memory usage in various drivers including XDR_STREAMING, IFRAME_STREAMING_POSTMESSAGE.
- ii. Transactions in queues are now supported.
- iii. HTTP origin checks (for CORS) are now case-insensitive.
- iv. XHR-based and XDR-based streaming responses are no longer stored in the debug log, as this was consuming excessive memory if debug logging was enabled.
- v. Speed up failover when connecting to a new node in a cluster.
- vi. Fixed an array-index-out-of-bounds error that occurred when a server is unreachable for a long enough period for the client to cycle through all drivers.

- vii. Try/catch block in EventDictionary's get() method has been replaced with code to test for existence of keys; this prevents browsers in debug mode "pause on all exceptions" from unnecessarily pausing in this method.
- viii. Fixed bug where DRIVER_CHANGE callback was invoked twice rather than once when a client switches to a new server in a cluster.
- ix. When moving to a new server in a cluster, we now re-attempt all configured drivers rather than expecting the driver used on the previous server to automatically be the appropriate choice for the second server.
- x. Fixed bug where a disconnected client could queue up multiple "session start" requests.
- xi. Support for event attributes to have values of "0".
- xii. Reconnection process for streaming drivers is now much faster.
- xiii. Certain chrome extensions use the postMessage API; these messages use an origin that our library did not expect – we now handle these and discard them rather than throw an error.
- xiv. Bug fix: JSON parsing error when receiving registered events has been fixed.

Nirvana 7.0 SP 2, Fix 1, Build 7.0.2.1.12094

Bug Fixes and Enhancements

Server

- i. Resolve issue where using certain methods to update the Realm ACL table (particularly when using the enterprise manager or setRealmACL methods) would fail when a WebSocket client was in the middle of the connection phase.
- ii. Reduce number of acknowledgements sent across the cluster when performing purge operations.
- iii. Added a cache refresh in nDataGroupEngine to fix caching problem on server as delta events were being incorrectly sent as snapshots

Nirvana 7.0 SP 2, Build 7.0.2.11880

Bug Fixes And Enhancements

Server

- i. Producers on the server will now yield CPU time to other threads while waiting to submit tasks to the Fanout executor. This will improve system load when large numbers of concurrent clients are publishing messages to the realm.
- ii. The buffer for reading events stored on disk has now been set to the same size as the maximum buffer size for reading published events from the network
- iii. Improve usage of heap memory when using multiplexed clients
- iv. Improve concurrent handling of closing a JavaScript streaming driver on the server.
- v. Fix reported publish count for channels, previously it would report double the number of events published.
- vi. Increase logging around authentication of users

C#

- i. Provide more detailed exception messages on various nDataGroup methods.

Java

- i. JMS Sessions will now correctly derive whether they are multiplexed from the Connection Factory.
- ii. Improve lookup of channels from the local client cache.

JavaScript

- i. The JavaScript client will now use a tick-based timer to schedule events. This should improve the handling of tasks such as disconnection detection in cases where the single thread is busy with other tasks (such as rendering GUI objects).
- ii. Fix transparent reconnections generating CLIENT_CLOSE callbacks in some scenarios.

Administration API

- i. Can no longer create ACL entries with an empty user or hostname
- ii. The datagroup tree will now correctly render groups with names that differ only by a "/" prefix.

Enterprise Manager

- i. Improve behaviour of the channel ACL table in enterprise manager

Nirvana 7.0 SP 1, Build 7.0.1.0.11750

Bug Fixes And Enhancements

Server

- i. Additional Logging for sessions using Cookies.
- ii. Additional auditing for DataGroups.
- iii. Improve handling of attempted creation of cyclic DataGroup memberships.
- iv. Fix to allow "." in queue names.
- v. Fix to always enforce lowercase realm names.
- vi. Increased logging of cluster lattice.
- vii. Allow Realm ACL to be modified without full permissions

Installer

- i. Fix potential issue with installer overwriting data directories on re-install.

Java Client API

- i. Stop rolled back events potentially being delivered to a consumer which is being closed.
- ii. Fix potential thread safety issue on Datagroup iterators

Administration API

- i. Updated XML import/export to include shm and multicast. – (Java & .Net)
- ii. .Net Admin API, resolved 2 reported issues with nRealmAdmin around creation of channels & application of ACLs.
- iii. Allow an nRealmNode to initialize without Data Group permissions (Java)

Enterprise Manager

- i. Fix Send and receive buffer size text on interface panel

JavaScript Client API

- i. Improve functionality in Safari 5 of XHR_STREAMING_CORS and XHR_LONGPOLL_CORS drivers.
- ii. Session initialization errors now correctly invoke the session's error callback listener (if the developer has assigned one).
- iii. User agent information now automatically shown in the debug log.
- iv. Improve failover between realms when first connection attempt to a cluster member fails
- v. Fix to prevent "freed script" errors when reloading a page that was using IFRAME based drivers in IE8/9.
- vi. Fix for XHR streaming where onerror is called more than once that affected restarting connectivity.

Nirvana 7.0.0, Build 7.0.0.0.11318

Bug Fixes And Enhancements

Server

- i. Resolve the cleanup of events on a channel / topic when using the JMS Engine for fanout to synchronous consumers. There was an issue where synchronous consumers connected to a topic would not be removed correctly resulting in events being stored but never cleaned up once consumed.
- ii. Improve performance of named object handling to use a faster collection for lookup and collection iterations.
- iii. Improve performance of synchronous channel consumers by enabling a batch / window size to be specified enabling faster delivery to client connections.
- iv. Resolve issue where cluster to cluster joins were not correctly handled in some situations where the join destination resource name exists on the source cluster.
- v. Improve nested group check during addStream calls so that only parent group arrays are checked. This issue caused additional time to be unnecessarily spent looping through data groups, which now does not happen.
- vi. Resolve resource leak when using data group listeners, where the listeners were not being correctly removed from data groups.
- vii. Ensure channel purge operations trigger status updates to keep admin APIs updated.
- viii. Fix a data group bug that could result in streams being incorrectly removed from a parent group when removed from a child group.
- ix. Fix a REST plugin bug where certain attributes in XML representations sent for publishing or purging events were always required and could not be omitted.
- x. Fix for batched create / delete of channels and queues when in a cluster. Depending on which node in the cluster the operation was performed, the stores were not always created or deleted on all realms.
- xi. Fix possibility of null entries being placed into an internal structure during data group flow control.
- xii. Ensure the default data group cannot be removed.
- xiii. Fix for recovery of events where incorrect membership could be reported from the channel subscriptions list resulting in events not being delivered correctly.
- xiv. Fix incorrect counter values for events consumed & received.
- xv. Resolve an issue that caused the setSubscriberName functionality to be lost during cluster event passing, resulting in all subscribers receiving the event from slave nodes.

- xvi. Identified and resolved an issue on Solaris/SPARC that was caused by the use of a high performance SpinLock mechanism that was problematic on SPARC architecture.
- xvii. Ensure snapshots for resources are delivered to new Admin API clients on connection to ensure information is up to date and correct.
- xviii. Resolve REST / SOAP / XML plugin delayed responses.
- xix. Resolve SOAP responses so that when a non standard port is used, this is reflected in the server response.
- xx. Resolve a thread synchronisation issue during inter realm disconnect that could prevent reconnection occurring.

Server / Java Client

- i. Fixed an issue with our internal dictionary. It was possible for a key to falsely match an existing key and therefore change the value associated with the existing key.
- ii. Ensure batched deletions remove channels from the internal cache.
- iii. Correctly enable/disable updates for listeners attached to the default datagroup

Admin API

- i. Resolve issue where status updates were not handled when the flag was set to not send all data groups on initial connection. Caused Admin API clients to not receive status updates for data groups with this flag set.
- ii. Resolved an issue that resulted in clusters not being connected to correctly.
- iii. Fix timing issue on reconnected connections during the setting up of listeners that could cause notifications to be missed.
- iv. nLinkNodes are now correctly used for realms with a mounted namespace.
- v. Fixed issue with nRealm object reuse which could cause non-mounted nRealmNode objects to appear mounted.
- vi. Fixed issue with nRealmNode reuse in joins or mountpoints where original nRealmNode is closed.
- vii. Fix setting Interface attributes through the Admin API.
- viii. Ensure all federated realms show up in the known realms list.

Enterprise Manager

- i. Resolve issue where status updates were not handled when the flag was set to not send all data groups on initial data stream connection. Caused Admin API clients to not receive status updates for data groups with this flag set.
- ii. Fixed sorting of tables where sorting is enabled.

- iii. Performance improvements on startup and in general.
- iv. Log streaming modified to use the Save dialog enabling creation of folders.
- v. Fix interface delete dialog where closing the delete dialog still removed the interface.
- vi. Fix channel connections where if a user disconnected and reconnected, the connection would not reappear on the channel connections table.
- vii. Ensure when realms are disconnected, the node is displayed correctly as disconnected in the tree without the node being clicked on.

.Net / Silverlight

- i. Fix asynchronous socket error handling so any async exceptions from the socket are handled and passed back through to the calling client code.
- ii. Expose the socket send / receive buffer sizes to provide better throughput if required.
- iii. Implement .Net multicast ack modes.

C++

- i. Implement encryption key rotation in multicast communication.
- ii. Upgraded Poco to 1.4.3p1 to resolve poco bug on windows XP 32bit which meant it would not run.
- iii. C++/IOS resolve potential null pointer on session reconnection.
- iv. Added ability to add a log listener
- v. Resolve potential missed events or duplicates in multicast
- vi. Fix the string representation of fSubject

JavaScript

- i. Fixed chunking protocols which allowed stream corruption (XDR Streaming and XHR CORS Streaming).
- ii. Performance improvements on all protocols.
- iii. Fixed latency issues with initial connections.
- iv. Fixed IE8 post message drivers issue.
- v. Removal of EA2 string from version comment.
- vi. Modified getData, setData so they are consistent with different variations of payload data.
- vii. Fix handling of byte arrays
- viii. Fix to allow websocket connections through reverse proxies that support it.
- ix. Fix to allow channel/queue names to be case-sensitive and include underscores
- x. Make session initialisation timeout a user-configurable value
- xi. Bug fix - batch re-subscription now uses the correct last eid.

- xii. Resolve various issues with `session.stop()`.
- xiii. Ensure a session, once stopped with `stop()`, can be started again with `.start()`.
- xiv. If a session is in a stopped or non-started state, then developer code can create a new session with a different configuration.
- xv. If a session start fails because of a cross-domain server configuration setting, failover to another driver is now much faster.
- xvi. Fix an issue where stopped sessions that used the `JSONP_LONGPOLL` driver may receive old events from a presumed timed out response.
- xvii. Improved the `XDR_STREAMING` driver to handle bugs in some versions of IE9.
- xviii. Fix an issue with sessions not being startable when `nirvana.js` is loaded dynamically *after* the containing page has already loaded.
- xix. Improved low level debug output.

Nirvana 7.0 Feature Summary

This release is the first production release of Nirvana 7.0. It contains all new features that were included in both EA1 and EA2, as well as additional improvements and features since the EA2 release. A quick summary of what is included in 7.0 is shown in the list below. Details of these can be found further down in this document.

- New ultra low latency Multicast functionality for inter-server cluster communications and Data Group client server communications.
- Shared memory driver support for ultra fast intra host
- Full support for the MQTT messaging standard.
- HTML5 JavaScript API re-engineering & entirely new API
- Improved performance giving lower latencies and higher throughput with updates to data group and channel fanout engines.
- Enhancements to data routing between Nirvana clusters
- Lightweight Synchronous Administration API
- Improved batching client API features

Nirvana 7.0 EA2 (11091) to Nirvana 7.0 Release

Enhancements

The release provides a number of changes including feature additions and performance improvements. The three main areas of change are:

- Nirvana Server
- Nirvana Enterprise Manager
- Nirvana Javascript API

Nirvana Server

The Nirvana server has been updated so that a lock free executor pattern is now in use for channels and datagroups, providing a highly optimized fanout mechanism for event delivery. The same pattern is now also used to enable protocol request pipelining through multiple executors resulting in performance improvements and efficiency improvements in the server's protocol processing modules. The new lock free executor comes with a configurable wait strategy to suit your appropriate deployment environment. At the moment, Nirvana provides spin, micro wait or block strategies.

Quite a few bugs have been fixed in the inter-cluster join functionality resulting in a reliability improvement and consistent behavior across joined clusters. Please note that this release will only allow cluster-wide to cluster-wide and local to local stores to be joined together. Using different store types will be supported in future releases.

The Nirvana server and Admin API have been modified to use hidden system datagroups to deliver status updates. This means that if a multicast configuration exists for the interface used to monitor / administer a Nirvana server, it can be automatically used hugely reducing the server overhead for delivering status updates. Additionally, the SHM driver has been optimized for performance and the Admin API has been enriched with GC statistics.

Finally, the Nirvana server has undergone optimizations in several areas such as String to Byte conversion, HTTP header parsing and ACL storage and management. All of these also result in reduced object creation, smaller memory footprint and therefore reduced GC calls and overall latency.

Enterprise Manager

The Nirvana Enterprise Manager has received several bug fixes and enhancements, specifically in areas where tree controls are being used such as the namespace tree and the JNDI tree. Both these trees now present nodes alphabetically sorted while pressing the * key toggles between expand / collapse of the currently selected sub tree.

Additionally, the namespace tree now supports rendering of a filtered subset so that users can only display a subset of the namespace that is relevant to their application, a great enhancement for Nirvana servers used for multiple applications. It is also possible to select multiple nodes of the same type in order to delete them in one click.

Finally, the JNDI tree now supports XA Connection factories and automatically expands sub trees.

Javascript API

The following new features have been added to the Nirvana Javascript API:

- New transport drivers have been added (EventSource and IFrame Streaming with postMessage proxying).
- Added ability to programmatically check if a driver is supported in the current browser runtime.
- Enhanced session creation configuration object with additional parameters (see API documentation for details).

- Removed support for P2P to align API with focus on channels, queues and datagroups.
- AlreadySubscribed exceptions now contain the resource that caused them to be raised.

The following performance improvements have been added to the Nirvana Javascript API:

- Improved DOM memory footprint
- Introduced stricter comparison / type checks throughout the API
- HTTP chunked encoding used where appropriate in all transport drivers (where this is supported)
- Changed HTTP GET methods to HTTP POST wherever was more appropriate
- No longer rely on cookies for session management
- Dynamically load JSON parsing code only if browser does not support JSON
- Modification of timeout logic to improve failover between transport drivers

The following security improvements have been added to the Nirvana Javascript API:

- Processing of postMessage requests is only allowed if they originate from a permitted domain
- Publishing to a channel without a valid ACL now correctly throws a SecurityException

The following bugs have been fixed in the Nirvana Javascript API:

- Correctly detect CTRL-ALT-D debug window functionality for Internet Explorer browsers
- Fixed an issue with subscribing to channels that were in the root of the namespace (not in a container)
- Fixed an issue with encoding message selectors correctly
- Fixed an issue with LP drivers where upon a session restart, multiple requests were made to the server

Nirvana 7.0 EA1 (10869) to Nirvana 7.0 EA2

This release is the 2nd Early Access version of Nirvana 7.0. As well as fixing minor bugs there is also some major additions between EA1 and EA2:

- New shared memory driver for intra-host messaging and improved performance.

- New data group fanout engine.
- New JavaScript API.
- Enterprise Manager update and reorganisation.

Shared Memory

With the introduction of the Nirvana Shared Memory Driver, client applications are able to specify a new RNAME protocol that will communicate with the Nirvana realm running on the same physical host through the physical machine therefore bypassing all Network IO. Clients connect to the Nirvana realm using the RNAME : shm://localhost/dev/shm for example where /dev/shm provides direct memory access.

Data Group Fanout

The fanout of events to data groups has been updated to remove any contention, thus improving latency further still. New internal processing flows and object structures have been written which enable Nirvana servers to deliver more events per second and with even lower latencies.

New JavaScript API

A new JavaScript API has been developed. This provides simplified session creation; the ability to attach multiple handlers to any callback; support for multiple transport drivers including XHR streaming, JSONP (in addition to existing WebSocket, forever iFrame and XHR long polling). In addition, there is a new Nirvana namespace providing access to a Nirvana Session and its factory methods for accessing resources such as channels, queues or services.

Enterprise Manager

The Enterprise Manager has been updated with a new look and feel and better icons. It has also had a slight reorganisation where the monitoring based panels have been moved into a separate 'Monitoring' panel. There have also been new panels added for configuration of shared memory and inter cluster connectivity. Lastly, the 'Interfaces', 'Multicast' and 'Shared Memory' configuration panels now exist in their own 'Comms' panel.

Nirvana 7.0 Early Access 1 (EA1) – What’s New

This release is an Early Access version of Nirvana 7.0. Version 7.0 introduces some new and exciting features and improvements, summarized below:

- New ultra low latency Multicast functionality for inter-server cluster communications and Data Group client server communications.
- Full support for the MQTT messaging standard.
- HTML5 JavaScript API re-engineering
- Improved performance giving lower latencies and higher throughput.
- Enhancements to data routing between Nirvana clusters
- Lightweight Synchronous Administration API
- Improved batching client API features

Multicast Support

With each major release we aim to further improve the performance of Nirvana both in terms of latency and throughput. Whilst our TCP Unicast delivery has once again been improved in 7.0, we have also introduced TCP Multicast delivery for inter-server cluster communications as well as for delivery via Data Groups. Multicast delivery to clients via Data Groups is transparent to the user and will automatically begin if the client is capable of receiving Multicast packets. No changes to the client application are necessary to begin using multicast.

This has enabled us to further drive down latencies to under 50 microseconds for many thousands of clients connected.

MQTT Support

MQTT (MQ Telemetry Transport), is a simple and lightweight messaging protocol offering publish/subscribe functionality. MQTT has been specifically designed for constrained devices deployed with access to minimal bandwidth. These design criteria make MQTT the ideal protocol for the rapidly growing “machine to machine” (M2M) or “Internet of Things” world of connected devices.

The protocol is gaining a lot of traction in the marketplace and is highly suited to embedded hardware platforms as well as mobile applications where bandwidth and battery power are at a premium.

Nirvana 7 provides transparent support for MQTT as a new delivery channel within our Nirvana realm server. This feature allows compliant MQTT (version

3.1) clients to seamlessly communicate with Nirvana and its wide range of supported Enterprise, Web and Mobile devices.

HTML 5 JavaScript

HTML5 JavaScript messaging is a major area of focus for Nirvana 7. Here is a brief summary of the changes that have been made:

- Wire protocol optimisation from client to server
- Streamlining of event processing within the client libraries
- Internal restructuring of libraries so that future HTML5 related enhancements can be added transparently

Performance Improvements

We have made significant progress again in terms of improving latency and throughput in Nirvana 7. In addition to the introduction of Multicast support we have also found additional gains in terms of our Unicast messaging. More can be found on the [benchmarking](#) section on our website.

Data Routing Between Nirvana Clusters

We have added the ability to use Nirvana joins to route data between Nirvana clusters. This feature is particularly useful for propagating data between Nirvana clusters over a WAN on a per channel basis.

Nirvana Admin API Enhancements

The Nirvana administration API now supports a synchronous 'configuration' only object which can be used to update Nirvana servers without the overhead of receiving asynchronous server monitoring updates.

Nirvana Client API Enhancements

We have extended support for batching of Nirvana API calls. Multiple channel creation and deletion operations can now be batched in single server calls.

Legal notices

Copyright © – Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.