



**GEMFIRE**<sup>®</sup>  
ENTERPRISE

*Release Notes*

Version 6.0

April 2009

Send comments on this manual to [docs@gemstone.com](mailto:docs@gemstone.com)

---

## INTELLECTUAL PROPERTY OWNERSHIP

This documentation is furnished for informational use only and is subject to change without notice. GemStone Systems, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this documentation.

This documentation, or any part of it, may not be reproduced, displayed, photocopied, transmitted, or otherwise copied in any form or by any means now known or later developed, such as electronic, optical, or mechanical means, without express written authorization from GemStone Systems, Inc.

Warning: This computer program and its documentation are protected by copyright law and international treaties. Any unauthorized copying or distribution of this program, its documentation, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted under the maximum extent possible under the law.

The software installed in accordance with this documentation is copyrighted and licensed by GemStone Systems, Inc. under separate license agreement. This software may only be used pursuant to the terms and conditions of such license agreement. Any other use may be a violation of law.

Use, duplication, or disclosure by the Government is subject to restrictions set forth in the Commercial Software - Restricted Rights clause at 52.227-19 of the Federal Acquisitions Regulations (48 CFR 52.227-19) except that the government agency shall not have the right to disclose this software to support service contractors or their subcontractors without the prior written consent of GemStone Systems, Inc.

This software is provided by GemStone Systems, Inc. and contributors "as is" and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall GemStone Systems, Inc. or any contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

## COPYRIGHTS

This software product, its documentation, and its user interface Copyright © 1997-2009, GemStone Systems, Inc. All Rights Reserved by GemStone Systems Inc.

JavaGroups Copyright 1999-2004 Free Software Foundation, Inc.

The Java Software technologies are Copyright © 1994-2000 Sun Microsystems, Inc. All rights reserved

GNU Trove copyright 2001-2004 Eric D. Friedman. The PrimeFinder and HashFunctions classes in Trove are copyright 1999 CERN - European Organization for Nuclear Research. Copyright © 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

MX4J Copyright © 2001-2004 by the MX4J contributors. All rights reserved.

JDBM Copyright © 2000 Cees de Groot. All Rights Reserved. Contributions are Copyright © 2000 by their associated contributors.

Antlr Copyright © 2005, Terence Parr. All rights reserved.

Commons Modeler Copyright © 2004 Commons Modeler. All rights reserved.

Trove Log4J Copyright © 1999 The Apache Software Foundation. All rights reserved. The Trove library is licensed under the Lesser GNU Public License, which is included with the distribution in a file called LICENSE.txt. PrimeFinder and HashFunctions classes in Trove © Copyright 1999 CERN - European Organization for Nuclear Research.

Copyright (C) 2006, Hitachi, Ltd. All Rights Reserved.

Copyright © 1994 Hewlett-Packard Company

Copyright © 1996,97 Silicon Graphics Computer Systems, Inc. Copyright © 1997 Moscow Center for SPARC Technology.

Copyright © 1998-2003 Daniel Veillard. All rights reserved.

Jgroups © 2001, 2002 www.jgroups.org

## PATENTS

GemFire is protected by U.S. patent 6,360,219. Additional patents pending.

## TRADEMARKS

GemStone, GemFire, GemFire Enterprise, and the GemStone logo are trademarks or registered trademarks of GemStone Systems, Inc. in the United States and other countries (trademark application pending for GemFire).

UNIX is a registered trademark of The Open Group in the U. S. and other countries.

Linux is a registered trademark of Linus Torvalds.

Red Hat and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

SUSE is a registered trademark of SUSE AG.

Sun, Sun Microsystems, Solaris, Forte, Java, Java Runtime Edition, JRE, and other Java-related marks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Intel and Pentium are registered trademarks of Intel Corporation in the United States and other countries.

Microsoft, Windows, and Visual C++ are registered trademarks of Microsoft Corporation in the United States and other countries.

IBM, AIX, and developerWorks are registered trademarks of IBM Corporation.

W3C is a registered trademark of the World Wide Web Consortium.

Other company or product names mentioned herein may be trademarks or registered trademarks of their respective owners. Trademark specifications are subject to change without notice. All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized to the best of our knowledge; however, GemStone cannot attest to the accuracy of all trademark information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

**GemStone Systems, Inc.**  
1260 NW Waterhouse Avenue, Suite 200  
Beaverton, OR 97006

---

These release notes cover new features and product changes in GemFire Enterprise and how to upgrade to this version of the product. They also list known issues.

For detailed information on using GemFire Enterprise to develop applications, please refer to the programming guides and programming API documentation included with the product. For information on administering GemFire Enterprise systems, see the *GemFire Enterprise System Administrator's Guide*. See the *GemFire Enterprise Developer's Guide* for information on programming with GemFire Enterprise.

This document contains the following sections:

- ▶ [New Features and Product Changes in 6.0 \(page 4\)](#)
- ▶ [Upgrading to Version 6.0 \(page 8\)](#)
- ▶ [Capacity Planning Implementation Note \(page 9\)](#)
- ▶ [Client/Server Implementation Note \(page 9\)](#)
- ▶ [Known Issues / Other \(page 10\)](#)

To contact GemStone Technical Support:

- ▶ [on the web: http://techsupport.gemstone.com](http://techsupport.gemstone.com)
- ▶ [by e-mail: techsupport@gemstone.com](mailto:techsupport@gemstone.com)
- ▶ [by phone: 800/243-4772 or 503/533-3503](tel:8002434772)

---

# New Features and Product Changes in 6.0

## Function Execution Service

The GemFire function execution service lets you execute application functions on a single member, in parallel on a subset of members, or in parallel on all members of a distributed system. Achieving linear scalability requires horizontal partitioning of application data so operations by distributed applications can be done concurrently across partitions. So, if the application requirements for transactions can be restricted to a single partition, and all data required for the transaction can be colocated to a single member or a small subset of members, then true parallelism can be achieved.

The term we use to describe this routing of application code to the data of interest is called *data-aware function routing*, or *behavior routing*.

For more information, see chapter 15, *Function Execution*, of the *GemFire Enterprise Developer's Guide*.

## Entry Eviction, Overflow, and Expiration for Partitioned Regions

Partitioned regions now support eviction, disk overflow, and expiration.

For information, see chapter 6, *Managing Data*, of the *GemFire Enterprise Developer's Guide*.

## Rebalancing and Redundancy Recovery for Partitioned Regions

Data in partitioned regions can now be migrated with partitioned region rebalancing. Rebalancing moves buckets to maintain a fair balance of data and behavior distribution across the distributed system. You manually invoke a rebalance operation on a specific member's cache through the GemFire API. Rebalancing can be used to add capacity to or remove capacity from an existing partitioned region, or to reduce variations in load due to uneven hashing of data. For more information, see section 6.6, *Rebalancing Partitioned Regions*, of the *GemFire Enterprise Developer's Guide*.

The default behavior for redundancy recovery is changed for partitioned regions. Partitioned regions will not recover redundancy until a new member is added to replace a member that has been lost. This behavior can be configured using the `recovery-delay` attribute. For more information, see section 11.3, *Data Redundancy and Storage*, of the *GemFire Enterprise Developer's Guide*.

## GemFire Resource Manager

The new GemFire resource manager works with your VM's tenured garbage collection (GC) to control heap use and protect your VM from hangs and crashes due to memory overload. The manager prevents the cache from consuming too much memory by evicting old data and, if the collector is unable to keep up, by blocking additions to the cache until the collector has freed an adequate amount of memory.

The resource manager has two settings, each expressed as a percentage of the total tenured heap, and both of which are disabled by default.

- ▶ **Eviction threshold.** Above this, the manager orders evictions for all regions with `eviction-attributes` set to `lru-heap-percentage`. This prompts dedicated evictions, independent of any application threads and it tells all application threads adding data to the regions to evict at least as much data as they add. Your VM's garbage collector removes the evicted data, reducing heap use. The evictions continue until the manager determines that heap use is again below the eviction threshold.
- ▶ **Critical threshold.** Above this, all activity that might add data to the cache is refused. This threshold is set above the eviction threshold and is intended to allow the eviction and GC work to catch up.

---

This VM, all other VMs in the distributed system, and all clients to the system receive `LowMemoryException` for operations that would add to this critical member's heap consumption. Activities that fetch or reduce data are allowed. For a list of refused operations, see the Javadocs for `ResourceManager.setCriticalHeapPercentage`.

For more information, see section 6.7, *Controlling Heap Use With the Resource Manager*, of the *GemFire Enterprise Developer's Guide*.

## Change to heapLRU Eviction Implementation

If you use heap LRU evictions in your pre-6.0 code, you must change the implementation for compatibility with 6.0.

The *GemFire Resource Manager* now monitors heap use and controls when heap LRU evictions are performed, so this work is not done at the region level. Entry evictions and total memory used evictions still operate the same.

The heap percentage, which in prior versions was set for each region in the `eviction-attributes`, has been moved into a cache-wide heap percentage that is set in the `resource-manager eviction-threshold`.

You still indicate that a region participates in heap LRU evictions by setting the `region-attributes eviction-attributes` to `heap-lru-percentage`.

Additionally, you can provide your own sizer class, with optional parameters, for region heap LRU evictions. This sizer is used by application threads adding to the region while evictions are in force. Each thread is required to remove at least as much data from the cache as it adds during heap evictions.

These are the changes to `eviction-attributes heap-lru-percentage` settings:

Pre-6.0 heap-lru-percentage setting	6.0 setting
<code>action</code>	No change.
<code>maximum</code> (for the region)	Moved into a maximum <i>for the cache</i> , set as the <code>cache resource-manager eviction-heap-percentage</code> attribute.
<code>time-interval</code>	No longer in use. The manager keeps track of the percentage in use for all regions. Before, if this interval wasn't set, the percentage was checked only on region modifications.
	<code>class-name</code> New <code>ObjectSizer</code> attribute. This serves the same purpose as the <code>ObjectSizer</code> class for the <code>lru-memory-size</code> eviction setting, which was available in prior releases. If you do not provide a sizer, GemFire uses its default sizer.
	<code>parameter</code> New, used to pass any parameters to the <code>ObjectSizer</code> identified in <code>class-name</code> .

For more information, see `eviction-attributes` in the region attributes list in section 4.5, *Region Attributes*, of the *GemFire Enterprise Developer's Guide*.

---

## Support for Backward Compatibility

GemFire 6.0 is backward compatible with version 5.7 for these configurations:

- ▶ Version 5.7 clients can connect to version 6.0 servers
- ▶ Version 5.7 and version 6.0 distributed systems can connect across the multisite (WAN) link

## Change to NOT\_AVAILABLE for CacheEvents and EntryEvents

If you use the constant `NOT_AVAILABLE` in your pre-6.0 code, you must change it for compatibility with 6.0.

Prior to version 6.0, `NOT_AVAILABLE` was used to indicate that an object value was not available in an event object. This applied to several situations including when a value would have been too expensive to obtain, like if an entry value was on disk instead of memory, or when an attribute was not available in certain contexts, like the callback argument on the remote side of a transaction.

In 6.0, generic typing was added to `Region`. With this change, because `NOT_AVAILABLE` is not an instance of the generic type `V`, returning it would have caused `ClassCastException`. So `null` is now used in the place of `NOT_AVAILABLE`.

These are the methods that now return `null` in cases where they used to return `NOT_AVAILABLE`:

- ▶ `CacheEvent.getCallbackArgument`
- ▶ `Region.put`
- ▶ `Region.destroy`
- ▶ `Region.remove`
- ▶ `Region.Entry.setValue`
- ▶ `EntryEvent.getOldValue`
- ▶ `EntryEvent.getNewValue`

As before 6.0, `null` is also returned when the value is just not there, like the old value for an entry creation. So this change has caused the `null` return value to have two possible meanings: “not available” and “no value.” To accommodate applications that must know which is intended, we have added the method `isOldValueAvailable` to `EntryEvent` and `isCallbackArgumentAvailable` to `CacheEvent`.

Make these changes to any code that you had before 6.0:

- ▶ Change all comparisons using `NOT_AVAILABLE` to compare to `null` instead.
- ▶ If you need to distinguish between “not available” and “no value”, when your comparison code returns `null`, disambiguate the response by calling the appropriate method `EntryEvent.isOldValueAvailable` or `CacheEvent.isCallbackArgumentAvailable`.

---

## Querying

This version includes improvements to querying and indexing to reduce memory overhead and improve the performance of query execution.

- ▶ Index evaluation is more intelligent. cutting down the generation of intermediate results wherever possible, reducing garbage generation and multiple index evaluation, speeding up query execution, and reducing the impact of the querying system on the server.
- ▶ In prior versions, query execution for a query that involved an index evaluation, additional constraints without indexes (constraints on fields that are not indexed), and projection columns were evaluated by first evaluating the index and returning the set of possible results to which additional constraints and projection attribute evaluation were performed subsequently. Now, the index evaluation phase uses a cost-based optimizer that applies additional constraints and evaluates projection attributes for better query execution times and a smaller footprint for the execution.
- ▶ Indexes have been simplified. Also, their implementation with respect to adding and removing indexes have been made more efficient. This reduces memory overhead attributed to having indexes on a region.
- ▶ The querying subsystem has been improved to allow for more concurrency, thus improving overall throughput of queries.
- ▶ In prior versions, partitioned region queries would fail with inconsistent exceptions if the nodes that the queries were routed to failed while executing the query. In effect, PR Querying did not take advantage of the HA aspects of the system. Now PR querying automatically retries queries on failed nodes to return the full resultset to the application.
- ▶ The OQL engine now supports the use of the `LIMIT` clause allowing application to limit the number of results being returned to the application.
- ▶ Diagnostics in the querying system have been improved with new stats and changes to existing stats to better reflect the operations on the system.

## Declarative definition of DataSerializer

This version includes support for `cache.xml` declaration of a data serializer implementation. In prior versions you could add it only through the `DataSerializer` API. For more information on data serialization, see section 17.2, *Performance Controls*, of the *GemFire Enterprise Developer's Guide*.

## Documentation Changes

This release includes updated Javadocs and product manuals, *GemFire Enterprise System Administrator's Guide* and *GemFire Enterprise Developer's Guide*. The documentation is located on the GemStone support web site at <http://www.gemstone.com/docs/6.0.0/product/docs/> and can be accessed directly on the web site or through the index page in the GemFire `productDir/docs` directory, where `productDir` is the GemFire Enterprise installation directory.

---

## Upgrading to Version 6.0

The migration from GemFire Enterprise 5.7 to 6.0 may require minor programmatic changes. Any migration requires a brief downtime of your system because you cannot connect to a running system with a different version number. These instructions assume you have read these release notes and are familiar with the feature changes described here.

If you are using a Sun JDK 1.4.X, you should upgrade. GemFire is no longer supported on Sun JDK 1.4.X. For the latest information on supported platforms, see *System Requirements* in the *Product Installation* chapter of the *GemFire Enterprise System Administrator's Guide*.

### Upgrade Steps

Follow these steps to migrate your development and production systems to the new version of GemFire. It is strongly recommended that your system migrated to GemFire 6.0 be thoroughly tested before being moved into production.

1. Install GemFire Enterprise version 6.0. Instructions are in provided the installation chapter of the *GemFire Enterprise System Administrator's Guide*. GemFire Enterprise 6.0 is installed as a complete product, rather than as a modification to a prior version. Starting with version 5.7, the Java JRE runtime environment is not bundled with GemFire Enterprise, so if you previously used the bundled JRE you will need to install and configure an appropriate JDK or JRE to comply with GemFire requirements and your unique system needs.
2. Review the changes documented in [New Features and Product Changes in 6.0 on page 4](#). Make any changes to your programs required for you to migrate to version 6.0. You may need to make `cache.xml`, `gemfire.properties`, or API changes for compatibility with the new version or to retain your application behavior. Make full use of the updated online Java API documentation for your modifications.
3. Recompile your Java applications against the `gemfire.jar` in this version of the product.
4. Stop all members of the system running with the prior version.
5. Point all member sessions to the new installation of GemFire Enterprise. For information, see the installation chapter of the *GemFire Enterprise System Administrator's Guide* and the programming chapter of the *GemFire Enterprise Developer's Guide*.
6. Restart all system members according to your usual procedures.

---

## Capacity Planning Implementation Note

GemFire Enterprise provides access to data at in-memory speeds. It also provides reliable notifications, high availability of data and notifications, and highly performant synchronous and asynchronous persistence and overflow to disk. To provide these features, GemFire servers rely on using system memory to initialize the data in a server, to store and dispatch updates to clients, and to buffer disk updates before they are flushed periodically to disk in asynchronous mode. All of these activities consume transient memory in the system. Initializing a region from another node, providing redundancy for notifications until they are reliably dispatched to clients, and buffering disk updates before they are flushed to disk all consume finite amounts of transient memory, over and above the memory used for data storage.

While GemFire is very efficient in pooling and reclaiming memory when not in use, the application designers and architects must do capacity planning to ensure that the server never runs out of memory or experiences periods of low performance due to spikes in transient memory usage. The server can be started with more memory using the standard `-Xmx` and `-Xms` memory settings. For information on this, see *System Member Performance* in the chapter *Monitoring and Tuning the Distributed System* in the *GemFire Enterprise System Administrator's Guide*. Most of the parameters that dictate the use of transient memory in the system are governed through `java` command-line and GemFire Enterprise parameters that you can tune before your applications go into production. To optimize the performance of your system for your specific configuration and processing needs, your GemFire applications architects and designers should take into account the memory needs of your applications during the transition to the GemFire Enterprise data fabric. You can also engage GemFire solution architects to help with this process.

## Client/Server Implementation Note

New applications should use the pool-based implementation of the client/server topology. The pool-based implementation is a much more performant and flexible implementation than the earlier, static server list implementation. The pool makes it very easy to configure clients to run against a server system without requiring static server configuration.

---

## Known Issues / Other

This section lists known issues in the GemFire Enterprise product and its supported platforms.

### Multiple threads must not be allowed to destroy the same region

When multiple threads are destroying a region or aborting region creation, a race condition in the setting and checking of the region's `destroyed` status can cause the region mapping to be nulled out multiple times. This in turn can corrupt statistics and static data. To avoid this, write your code so that:

- ▶ `destroyRegion` acquires the `destroyLock`
- ▶ `close` and `localDestroyRegion` do not acquire the `destroyLock`

### Using IPv6: RMI does not allow server to bind to a link-local address

All IPv6 link-local addresses start with `fe80:*`. GemFire JMX agent binds to the local host unless an `rmi-bind-address` is provided as a system property, in the `gemfire.properties` file, or through the command line while start the agent. On many common configurations running Windows operating system, this causes an issue as `java.net.InetAddress.getLocalHost()` returns an IPv6 link-local address and the user cannot correct the problem easily. To work around the issue, the user can provide an `rmi-bind-address` explicitly.

On systems running Windows, use the command `ipconfig /all` tool to list all of the valid IP addresses.

For Linux/AIX/Solaris, use the command `ifconfig -a` tool to list all of the valid addresses for a machine.

The following command starts an agent and provides an RMI bind address so that it does not conflict with IP addresses used by the system:

```
.\agent.bat start rmi-bind-address=fdf0:76df:b0ed:9449::17
```

This issue occurs commonly on systems running Windows and rarely on systems running other operating systems.

### Starting the JMX agent specifying an incorrect path for the property file

When you start a JMX agent with an incorrect path for the property file, the agent starts and no error is reported.

### Starting a JMX agent while using locators

A GemFire JMX agent can be started either by passing the locator(s) information through the `locators` parameter or by using the multicast address and port number settings that you are using for the distributed system member discovery.

To start a JMX agent using locators, indicate the following parameters: `-dir`, `mcast-port=0`, `locators=locator1-host:locator1-port,locator2-host:locator2-port`.

`-dir` indicates the directory where the agent creates an `agent.ser` file to which it writes the agent status and saves to disk as a serialized object. If the agent is killed or stopped abruptly for some reason, ensure that that `*.ser` file is deleted before restarting the agent. The agent does not restart if the `*.ser` file is not deleted from disk.

To start an agent using locators:

```
%GEMFIRE%\bin\agent start -dir=.. mcast-port=0 locators=
localhost:37000,localhost:37001 log-file=../log/agent.log
```

---

To start an agent using multicast:

```
%GEMFIRE%\bin\agent start -dir=.. mcast-port=10339 mcast-address=
239.192.81.1 log-file=../log/agent.log
```

To stop a JMX agent, include the directory name in which the agent process was started and the \*.ser file was created:

```
%GEMFIRE%\bin\agent stop -dir=.\agent-dir
```

## Saving statistic alert definitions in a JMX admin agent

While making changes to the statistic alert definitions of a JMX agent, if the user does not have write permissions to the agent.ser file or to the directory that contains agent.ser file, a warning is displayed indicating that the user cannot save additional definitions to the file.

## Use locators only while using NIC bonding and GemFire

When using NIC-bonded Ethernet cards on a Linux platform, regardless of the NIC bonding mode, do not use multicast-based member discovery. NIC bonding on Linux platforms causes issues with GemFire multicast member discovery capability and members might not locate each other, especially during member start up.

To work around this issue, *always* use locator-based member discovery.

## GemFire is no longer supported on Sun JDK 1.4.X

For the latest information on supported platforms, see *System Requirements* in the *Product Installation* chapter of the *GemFire Enterprise System Administrator's Guide*.

## Compiling previously written code containing JDK 5 parameters

If you are currently implementing customized code that uses JDK5 parameterized types and you have upgraded to GemFire 6.0, then the code may no longer compile with GemFire 6.0

For example the following code compiles with GemFire 5.7 but not with GemFire 6.0:

```
{{{
  Set<Region> set = cache.rootRegions();
}}}
```

To fix this issue, make the following change in the code:

```
{{{
  Set<Region<?,?>> set = cache.rootRegions();
}}}
```

If the previously written code was written only using JDK4 features, then no change is required for it to compile with GemFire 6.0.

However, note that GemFire 6.0 is supported with JDK5.

## Weblogic server and UTF encoding

Some Weblogic servers are unable to process XML files with a UTF encoding specified. To disable the UTF encoding, make sure the cache.xml file specifies `<?xml version="1.0"?>` and not `<?xml version="1.0" encoding="UTF-8"?>`.

## Acrobat Reader 6.0 links always go to first page

There is a known problem with Acrobat Reader 6.0 that causes links between pdf files to always go to the first page. If you encounter this problem while viewing the documentation pdfs, you can try these

---

steps. Open Acrobat 6, open the `Edit` menu, select `Preferences`, click on `General`, then uncheck the "Open cross-document links in same window" option.