

The GemFire logo consists of the word "GEMFIRE" in a bold, black, sans-serif font, with a small orange triangle above the letter "I". A thin green horizontal line is positioned below the word. Below "GEMFIRE" is the word "ENTERPRISE" in a smaller, black, sans-serif font. The logo is centered within a large, light gray, downward-pointing triangle that is part of a larger pattern of similar triangles in the background.

GEMFIRE[®]
ENTERPRISE

Early Access Features

Version 5.5

March 2008

Send comments on this manual to 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 © 1997-2008, GemStone Systems, Inc. All Rights Reserved by GemStone Systems Inc.

Java Software technologies Copyright © 1994-2000 Sun Microsystems, Inc. All rights reserved. Sun Java Runtime Environment Copyright © 2004-2008 Sun Microsystems, Inc. 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.

ILOG JViews copyright © 1996-2001 ILOG S.A. All rights reserved.

JavaGroups copyright © 1999-2004 Free Software Foundation, Inc.

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 project (<http://mx4j.sourceforge.net>). Copyright © 2001-2004 by the MX4J contributors. All rights reserved.

Copyright © 2004 Commons Modeler. All rights reserved.

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

Copyright © 1993-95 Ioi Kim Lam. Copyright © 1996 Expert Interface Technologies.

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

Antr © 2005-2007, Terence Parr, Copyright © 2003-2007, Terence Parr All rights reserved.

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.

ILOG is a registered trademark and JViews is a trademark of ILOG S.A.

Exolab is a registered trademark of ExoLab Group.

IBM 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

This document describes the early access features included in this release.

The functionality documented here is included as part of an early access program, with specific, targeted customers in mind. It may not yet be fully implemented, and it has not been subject to the full set of rigorous testing procedures that our General Availability software receives. If you wish to use this feature prior to its General Availability, we highly advise you first inform your GemStone technical representative, so additional support can be provided and any new requirements can be considered for implementation into the full release.

GemStone Systems, Inc. provides these features without warranty and does not currently endorse its use in any critical applications.

For information on general access features provided with this release, please refer to the *GemFire Enterprise Release Notes* for version 5.5. 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 detailed information on administering GemFire Enterprise systems, see the *GemFire Enterprise System Administrator's Guide*. For detailed information on programming with GemFire Enterprise, see the *GemFire Enterprise Developer's Guide*.

GemFire Enterprise 5.5 includes these early access features:

- ▶ *Region Snapshots - Early Access* on page 4
- ▶ *Querying in Client/Server - Early Access* on page 5
- ▶ *EntryEvent Get Serialized Values - Early Access* on page 5
- ▶ *Load Balancing in Multi-site Installations - Early Access* on page 6

Region Snapshots - Early Access

Snapshots are managed through the `saveSnapshot` and `loadSnapshot` methods of the `Region` interface. These are the key considerations for snapshots:

- ▶ Taking a snapshot does not stop region activity, so modifications to the region that are made while the snapshot is being taken may or may not be saved to disk.
- ▶ Snapshots are not recursive. The region's entries are saved, but subregions and subregion entries are not.
- ▶ Loading a snapshot removes all existing distributed region contents before loading the data from disk. All existing subregions and entries are destroyed before the snapshot is loaded. Remote regions with the same name are cleared of all subregions and entries, and remote regions that are configured as replicas do a new `getInitialImage` operation to get the data from this snapshot. All existing references to the distributed region become unusable and, if used, throw a `RegionReinitializedException`. All applications with references to the region must reacquire the region using `Cache.getRegion` or `Region.getSubregion`.

For additional information on snapshots, see the online Java documentation.

Querying in Client/Server - Early Access

You can run an ad-hoc query on a server from a client. When you use `Region.query` on a region configured as a client region, the query executes on the server, and the results are returned to the client. To do join queries, you can use either region of the join.

Only queries run through the `Region.query` method are sent to the server. Queries run by the client through the `GemFire QueryService` execute against the client cache.

In client/server querying, the `Region.query` method can take any full query, or it can take a predicate as in peer-to-peer querying. Otherwise, except for being run against the server cache, client/server queries operate like peer-to-peer queries. For details on peer-to-peer querying, see *Querying and Indexing*, in the *GemFire Enterprise Developer's Guide*.

EntryEvent Get Serialized Values - Early Access

You can retrieve serialized values from `EntryEvent`. This is useful if you get values from one region's events just to put them into a separate cache region. The standard `getOldValue` and `getNewValue` functions deserialize the values, which would then be reserialized when you `put` them into the cache. There is no counterpart `put` function as the `put` recognizes that the value is serialized and bypasses the serialization step.

For details on the new API supporting this functionality, see the online Java documentation in `com.gemstone.gemfire.cache` for the interface `SerializedCacheValue` and for the `EntryEvent` methods `getSerializedNewValue` and `getSerializedOldValue`.

Load Balancing in Multi-site Installations - Early Access

GemFire Enterprise now allows you to balance the load of gateway communication between multiple VMs. The basic approach is to define multiple gateway hubs and then split outgoing region event messaging between them.

With the new functionality, you can:

- ▶ Specify multiple gateway-hubs in your cache
- ▶ Indicate for each VM whether a hub should start as primary or secondary
- ▶ Point each gateway-enabled region to a specific gateway-hub

Only the primary hub instances send region events to remote sites. By splitting your hub primaries between multiple VMs and splitting your region events between hubs, you can achieve a level of load balancing in your outgoing hub messaging.

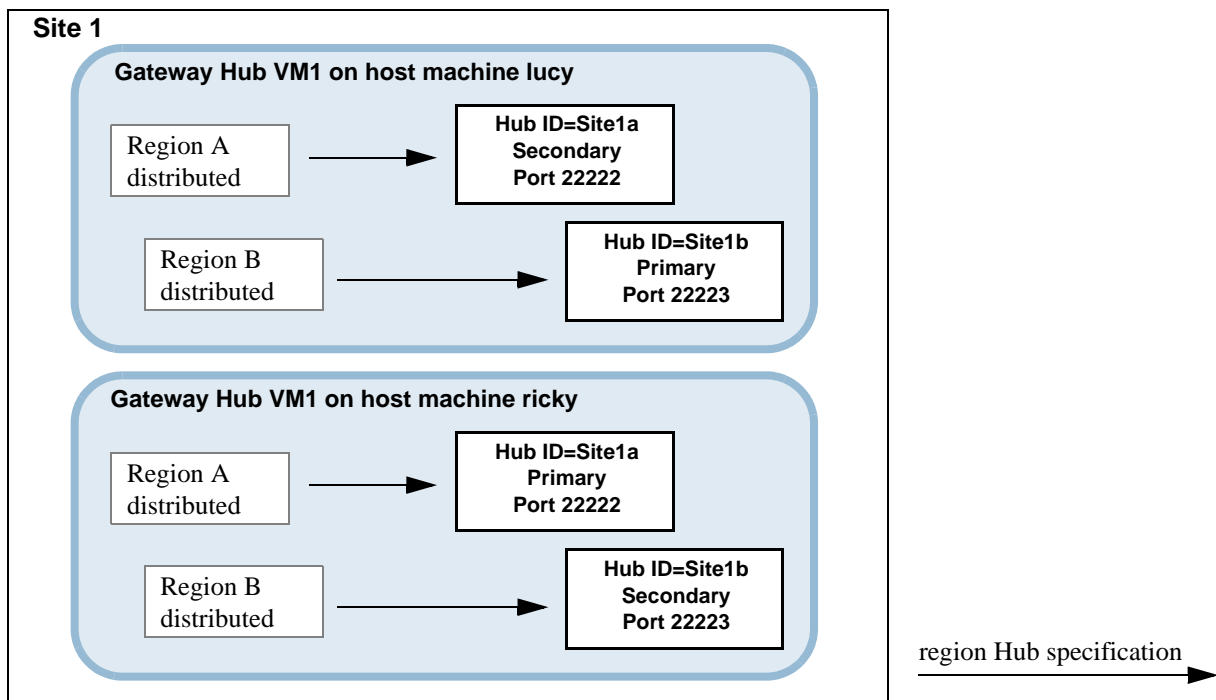
The Multiple-Hub Configuration

Figure 1.1 shows a typical multiple-hub configuration. There may be any number of VMs in this distributed system in addition to the ones that host the gateway hubs.

In this configuration:

- ▶ There are two VMs hosting gateway hubs.
- ▶ There are two gateway hubs in each hub VM. One hub is primary in VM1 and the other hub is primary in VM2. Only the primaries send data to remote sites.
- ▶ Each hub has some of the region traffic pointing to it.

Figure 1.1 Two-Hub Configuration With Primaries in Different VMs

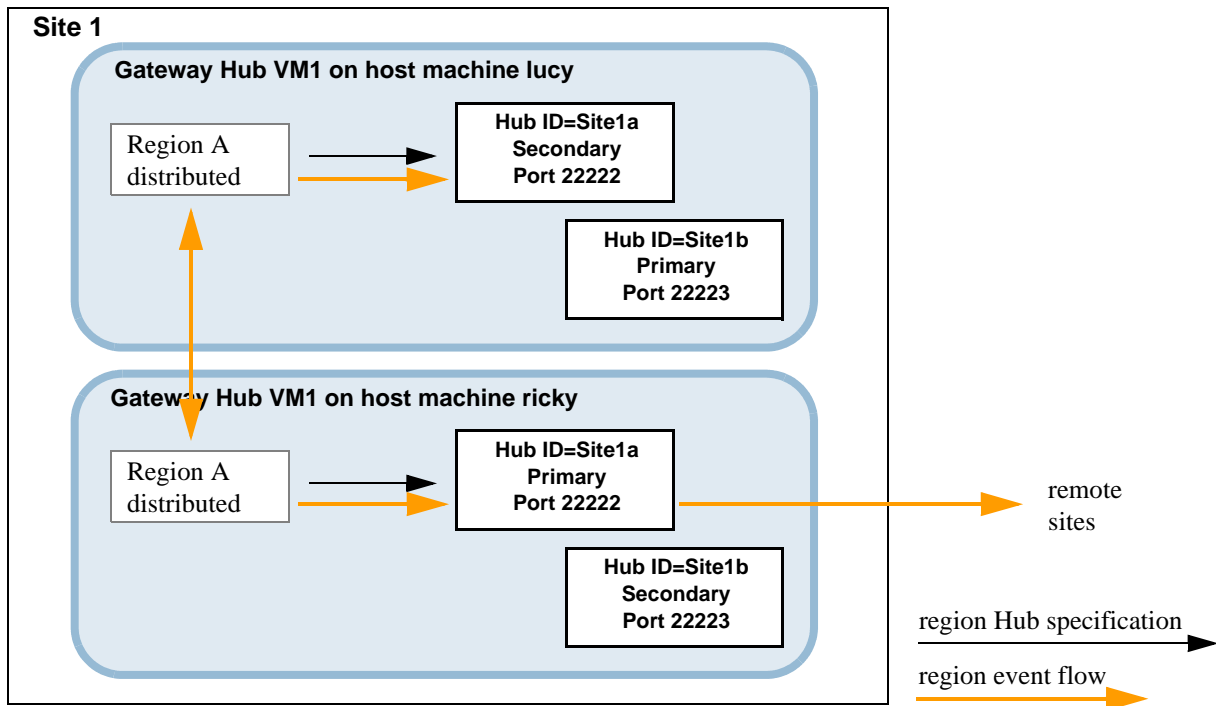


Event Flow With Multiple Hubs

Figure 1.2 shows event flow for region A in the multiple-hub configuration of Figure 1.1 on page 6.

- ▶ Region events originating in VM1 are stored in the local secondary Hub Site1a and are also distributed to VM2. Then in VM2, the events arriving in the local Region are sent to the VM's primary Hub Site1a, which then forwards them to remote sites.
- ▶ Region events originating in VM2 are sent directly out of the local primary Hub Site1a. They are also distributed to VM1 for cache update and storage in the secondary Hub Site1a.
- ▶ HubSite1b never sees this region's outgoing events.

Figure 1.2 Outgoing Region Event Flow for Two-Hub Configuration



If you have multiple hubs, make sure you explicitly assign one hub to each distributed region using the `hub-id` region attribute. If you do not specify a hub for a region, its events will go to all hubs and each hub's primary will send the message to the remote sites, resulting in duplicate sends.

Configuring the Multiple-Hub Site

These tables list the new configuration attributes for multi-site load balancing.

Table 1.1 Gateway Hub Attribute

gateway-hub attribute	Description	Default
startup-policy	Specifies whether the hub should attempt to start as primary or secondary hub. Valid values are <code>primary</code> , <code>secondary</code> , and <code>none</code> .	none

Table 1.2 Region Attribute

region attribute	Description	Default
hub-id	Specifies the destination hub for region events. <i>For multiple-hub systems this must be specified. If it is not, region events go to all available hubs, which results in duplicate sends to remote sites.</i>	" "

Hub Configuration

The `cache.xml` in this example defines the gateway hub configurations for VM1 in [Figure 1.1 on page 6](#).

Example 1.1 Two-Hub Gateway Hub Configuration in `cache.xml`

```
<cache>
  <gateway-hub id="Site1a" port="22222" startup-policy="secondary">
    ...
  </gateway-hub>
  <gateway-hub id="Site1b" port="22223" startup-policy="primary">
    ...
  </gateway-hub>
```

This shows the API equivalent.

Example 1.2 Two-Hub Gateway Hub Configuration Through the API

```
// Create or obtain the GemFire cache
Cache cache = ... ;

// Create the first Gateway Hub
GatewayHub site1aHub = cache.setGatewayHub("Site1a", 22222);
site1aHub.setStartupPolicy("STARTUP_POLICY_SECONDARY");

// Create the second Gateway Hub
GatewayHub site1bHub = cache.setGatewayHub("Site1b", 22223);
site1bHub.setStartupPolicy("STARTUP_POLICY_PRIMARY");
```

```
// Define gateways and endpoints for the gateway hubs ...
```

Region Configuration

When you use multiple hubs, you must point each gateway-enabled region at one of the hubs. If you do not specify a hub, region events are delivered to every hub and then sent to remote sites by each hub's primary instance. The region configuration must be consistent across the distributed system, even in caches where there are no hubs running.

Split region events between your hubs as evenly as you can. If you have two hubs and three regions, for example, with one region having the bulk of the events, put that region on one hub and the others on the other hub.

This `cache.xml` assigns the example regions in [Figure 1.1 on page 6](#) to their respective hubs.

Example 1.3 Enabling Region Communication With Gateway Hubs in `cache.xml`

```
<region name="A">
  <region-attributes ... enable-gateway="true" hub-id="Site1a"/>
  ...
</region>
<region name="B">
  <region-attributes ... enable-gateway="true" hub-id="Site1b"/>
  ...
</region>
```

This is the same configuration through the API.

Example 1.4 Enabling Region Communication With a Gateway Hub Through the API

```
// Region A
AttributesFactory factory = new AttributesFactory();
factory.setEnableGateway(new Boolean(true));
factory.setGatewayHubId("Site1a");
Region multiSiteRegionA = cache.createRegion("A", factory.create());

// Region B
AttributesFactory factory = new AttributesFactory();
factory.setEnableGateway(new Boolean(true));
factory.setGatewayHubId("Site1b");
Region multiSiteRegionB = cache.createRegion("B", factory.create());
```

How Hub Startup Policy Affects Startup Behavior

Every hub must have one primary instance running to maintain communication with remote sites. The other instances are secondaries that act as backups to the primary.

You can specify which hub instances should start as primary and which as secondary in your distributed system. When a hub is initialized, it attempts to assume the role specified, but will start up even if it must assume a different role.

This table describes startup behavior for the three policies.

Table 1.3 Gateway Hub Startup Policy and Startup Behavior

Startup Policy	Startup Behavior
none (default)	If no primary is running for the id, the hub starts as primary. Otherwise it starts as secondary.
primary	Same behavior as for none with the addition that the hub logs a warning if it starts as secondary.
secondary	If a primary is present, the hub starts as secondary. If there is no primary, the hub waits for up to one minute for a primary to start. If no primary starts in that time, the hub starts as primary and logs a warning.

Configuring Remote Sites to Point to Your Multiple-Hub System

The regions' hub specifications only affect which hub processes their outgoing events. For incoming events, hubs process every event received regardless of their destination regions. For high availability, specify all of your hubs as gateway endpoints on your remote site. For any event, the remote site will send to only one endpoint.

This is a valid remote gateway specification pointing to our example site.

Example 1.5 Remote Site Configuration Pointing to the Multiple-Hub Site

```
<cache>
  <gateway-hub id="Some remote gateway hub ...
    <gateway id="Sitela" ...
      <gateway-endpoint id="SitelaLucy" host="lucy" port="22222"/>
      <gateway-endpoint id="SitelaRicky" host="ricky" port="22222"/>
      <gateway-endpoint id="SitelbLucy" host="lucy" port="22223"/>
      <gateway-endpoint id="SitelbRicky" host="ricky" port="22223"/>
    </gateway>
  </gateway-hub>
```

Basic Configuration Steps

To implement a multiple-hub system, first determine the configuration of the connections between the multiple distributed system sites. Then, for each site, follow these steps:

1. Identify the VMs that will host your gateway hubs, including which machines they will run on the addresses they will use for gateway communication.
2. Decide the gateway hub startup policy for each VM's hubs.
3. Decide which regions will point to which hubs.
4. Following your planned configuration, for each hub define the gateways to the remote sites, including:
 - ▶ Endpoints - point to every hub instance on the remote site
 - ▶ Queue attributes
5. Enable regions to communicate updates to the gateways, specifying:
 - ▶ enable gateway
 - ▶ hub ID
6. Start all host VMs in the distributed system at the same time. This allows the primaries to assume their roles before any secondaries default to primary.

For details on the basic configuration of gateway communication, see the information on multi-site configuration in the *GemFire Enterprise Developer's Guide*.

