|  |
| --- |
| ИнтерТраст |
| Настройка глобального кэша |
|  |

|  |
| --- |
| Митавский Д. В. |

Содержание

[Настройка основных параметров 3](#_Toc449456754)

[Настройка кластерного режима работы 4](#_Toc449456755)

[server.properties 4](#_Toc449456756)

[JMS 4](#_Toc449456757)

[JBoss 7.1, EAP 6.2, Wildfly 8, 9: 4](#_Toc449456758)

[Wildfly 10: 5](#_Toc449456759)

# Настройка основных параметров

Основные параметры работы глобального кэша производятся в server.properties

1. Включить глобальный кэш. Внимание, если кэш уровня транзакции (кэш 1го уровня) отключен, то глобальный кэш (кэш 2го уровня) также работать не будет.

**cache.domainObject.enabled**=**true  
global.cache.enabled**=**true**

1. Размер кэша. Единица измерения: мегабайты (буква M) или гигабайты (буква G)

**global.cache.max.size**=**100M**

Обычно, не рекомендуется задавать размер кэша больше 2 гигабайт. Если в этом есть реальная необходимость, то совместно с таким кэшем нужно использовать сборщик мусора G1 (<http://www.oracle.com/technetwork/tutorials/tutorials-1876574.html>).

1. Режим работы глобального кэша. По умолчанию – блокирующий (blocking). Другой пока не реализован.

**global.cache.mode**=**blocking**

1. Сбор расширенной статистики. Требуется при мониторинге скорости работы глобального кэша. Обычно в этом нет необходимости.

**global.cache.extended.statistics**=**false**

1. Режим отладки. Его включение необходимо лишь в случае, когда есть подозрение некорректности работы глобального кэша. Если режим отладки включен, то каждое получение значения из кэша сопровождается запросом в базу, сверяющим результат.

**global.cache.debug.enabled**=**false**

# Настройка кластерного режима работы

## server.properties

Если сервера приложений объединены в кластер, то для синхронизации глобального кэша между узлами кластера необходимы следующие настройки в server.properties:

1. Включение режима работы в кластере:

**global.cache.cluster.mode**=**true**

1. Периодичность обновления узлов в кластере в **секундах** по сигналам, приходящим из других узлов. По умолчанию, синхронизация происходит сразу по получению сообщения; время доставки сообщения не фиксировано, сами сообщения асинхронны. В тех системах, где узлы кластера отведены под группы пользователей, слабо между собой взаимодействующих, время можно установить довольно большим (десятки минут), чтобы избежать ненужных сбросов кэшей коллекций и, соответственно, дополнительных запросов к базе данных.

**global.cache.cluster.synchronization.seconds**=**30**

## JMS

1. Необходимо произвести настройку JMS согласно документам \CM5\Инфра проекта\Настройка JMS…
2. В настройках JMS сервера приложений, необходимо добавить
3. Фабрику соединений с JNDI-именем **java:/GlobalCacheClusteredConnectionFactory**. Важно обеспечить распределёность сообщений, производимых фабрикой. В случае вышеуказанных документов, необходимо указать коннектор “http-connector” для Wildfly 10 или “netty” для JBoss 7.1/6.2 EAP/Wildfly 8, 9.
4. Фабрику соединений с JNDI-именем **java:/GlobalCacheLocalConnectionFactory**. Эту фабрику, наоборот, необходимо настроить так, чтобы её сообщения не выходили за рамки текущего сервера приложений, что соответствует коннектору “in-vm” из документации по настройке JMS.
5. «Тему» (Topic) с JNDI-именем: **topic/ClusterNotificationTopic**
6. «Очередь» (Queue) с JNDI именем **queue/ClusterNotificationDelayQueue**.

Примеры настройки фабрик, очереди и темы для различных серверов приложений.

### JBoss 7.1, EAP 6.2, Wildfly 8, 9:

<**connectors**>  
 <**netty-connector name="netty" socket-binding="messaging"**/>  
 <**netty-connector name="netty-throughput" socket-binding="messaging-throughput"**>  
 <**param key="batch-delay" value="50"**/>  
 </**netty-connector**>  
 <**in-vm-connector name="in-vm" server-id="0"**/>  
</**connectors**>  
<**acceptors**>  
 <**netty-acceptor name="netty" socket-binding="messaging"**/>  
 <**netty-acceptor name="netty-throughput" socket-binding="messaging-throughput"**>  
 <**param key="batch-delay" value="50"**/>  
 <**param key="direct-deliver" value="false"**/>  
 </**netty-acceptor**>  
 <**in-vm-acceptor name="in-vm" server-id="0"**/>  
</**acceptors**>

<**jms-connection-factories**>  
 ...  
 <**pooled-connection-factory name="GlobalCacheLocalConnectionFactory"**>  
 <**connectors**>  
 <**connector-ref connector-name="in-vm"**/>  
 </**connectors**>  
 <**entries**>  
 <**entry name="java:/GlobalCacheLocalConnectionFactory"**/>  
 <**entry name="java:jboss/exported/jms/GlobalCacheLocalConnectionFactory"**/>  
 </**entries**>  
 </**pooled-connection-factory**>  
 <**pooled-connection-factory name="GlobalCacheClusteredConnectionFactory"**>  
 <**transaction mode="xa"**/>  
 <**connectors**>  
 <**connector-ref connector-name="netty"**/>  
 </**connectors**>  
 <**entries**>  
 <**entry name="java:/GlobalCacheClusteredConnectionFactory"**/>  
 <**entry name="java:jboss/exported/jms/GlobalCacheClusteredConnectionFactory"**/>  
 </**entries**>  
 </**pooled-connection-factory**>  
 ...  
</**jms-connection-factories**>  
  
<**jms-destinations**>  
 ...  
 <**jms-topic name="ClusterNotificationTopic"**>  
 <**entry name="topic/ClusterNotificationTopic"**/>  
 </**jms-topic**>  
 <**jms-queue name="ClusterNotificationDelayQueue"**>  
 <**entry name="queue/ClusterNotificationDelayQueue"**/>  
 </**jms-queue**>  
 ...  
</**jms-destinations**>

### Wildfly 10:

<**http-connector name="http-connector" endpoint="http-acceptor" socket-binding="http"**/>  
<**in-vm-connector name="in-vm" server-id="0"**/>  
<**http-acceptor name="http-acceptor" http-listener="default"**/>  
<**in-vm-acceptor name="in-vm" server-id="0"**/>

<**jms-queue name="ClusterNotificationDelayQueue" durable="false" entries="queue/ClusterNotificationDelayQueue"**/>  
<**jms-topic name="ClusterNotificationTopic" entries="topic/ClusterNotificationTopic"**/>  
<**pooled-connection-factory name="GlobalCacheLocalConnectionFactory" max-pool-size="100" thread-pool-max-size="100" scheduled-thread-pool-max-size="20" entries="java:/GlobalCacheLocalConnectionFactory java:jboss/exported/jms/GlobalCacheLocalConnectionFactory" connectors="in-vm"**/>  
<**pooled-connection-factory name="GlobalCacheClusteredConnectionFactory" transaction="xa" entries="java:/GlobalCacheClusteredConnectionFactory java:jboss/exported/jms/GlobalCacheClusteredConnectionFactory" connectors="http-connector"**/>