



IBM Software Group

Best practices: setting up session replication in IBM WebSphere Application Server Version 6.1

Ashish Ghodasara

acghodas@us.ibm.com

WebSphere Application Server Technical Support



WebSphere® Support Technical Exchange



Agenda

- Understanding HttpSession
- Failover support
- Memory-to-Memory Replication
- Peer-to-Peer topology
- Client/Server topology
- Best practices and Tuning consideration
- Questions

HttpSession Object

- A Session is a series of requests to a servlet, originating from the same user at the same browser
- Allows to keep track of individual users by using unique session ID
- Session ID generated on first request and send back to browser with first response.
- Session ID then arrives with each subsequent request
- Cookies, URL rewriting and SSL (if request is on HTTPS) are ways to track the session IDs
- Example of web application using sessions is to store shopping cart information for user in session object

Developing session management in servlets

- Get the HttpSession object.
 - ▶ Use the getSession() method of the javax.servlet.http.HttpServletRequest object in the Java Servlet 2.4 API
 - ▶ With this method, webcontainer will give existing session if one exist, else give new one
- Store and retrieve user-defined data in the session.
 - ▶ Use setAttribute (java.lang.String name, java.lang.Object value) method to store name and value pair. You can use other methods from HttpSession Interface API
- Provide feedback to the user
 - ▶ As per your business logic
- Notify Listeners
 - ▶ If the Objects in HttpSession implements javax.servlet.http.HttpSessionBindingListener interface, then notification triggers before ending session to do post-session Processing
- End the session
 - ▶ Let WebSphere Session manager invalidate based on inactivity timeout set in session manager
 - ▶ Or use **invalidate()** method

Example code

```
<%  
  //obtaining session, this method call will return new session if one does not exist  
  HttpSession s = request.getSession();  
  
  Integer hits = new Integer(1);  
  // if this is new session  
  if (s.isNew())  
  {  
%>  
    This is New session, this is your 1st hit  
  <%  
    s.setAttribute("numberOfHits",hits);  
  }  
  //if this is existing session, increment counter and display  
  else  
  {  
    hits = (Integer) s.getAttribute("numberOfHits");  
    hits = new Integer (hits.intValue () + 1);  
    s.setAttribute("numberOfHits", hits);  
  }  
%>  
  This is existing session and you hit this page for <%=hits.toString()%> times  
  <%  
  }  
%>
```

Distributed sessions

- Session management can store session-related information in ...
 - ▶ In application server memory (the default). This information cannot be shared with other application servers.
 - ▶ In a database. This storage option is known as *database persistent sessions*.
 - ▶ In another WebSphere Application Server instance. This storage option is known as *memory-to-memory sessions*.

- Distributed sessions are essential for using HTTP sessions for the failover facility

- Plug-in maintains Session affinity by sending requests with session ID to cluster member where that session has been previously created

- On the failure of cluster member, plug-in routes requests to other members

- When cluster member receives a request associated with a session ID that it currently does not have in memory, it can obtain the required session state by accessing the external store (database or memory-to-memory).

Data Replication Service

- Data Replication Service (DRS) is an internal WebSphere® Application Server component that replicates data
- This service is available for Network Deployment or higher configuration
- Session Manager, Dynamic Cache Service and Stateful session bean are the consumers of DRS
- Configure DRS by creating a replication domain
- Replication Domain can be created either manually from Environment page of admin console or automatically at cluster create time

Memory-to-Memory Replication

- In Memory-to-Memory session replication, session data from owner server replicates (copies) in one or more other application server using Data Replication Service
- This requires creation of a data replication service instance in owner server that communicates to other data replication service instances in remote application servers
- This can be achieved by creating a replication domain and configuring the session manager of cluster members to use memory-to-memory replication
- A server can run in one of the following three modes:
 - ▶ Client mode: server only sends copies of sessions it owns
 - ▶ Server mode: server only stores backup copies of other servers
 - ▶ Both mode: server sends copies of session it owns and acts as a backup table for sessions owned by other application server
- Two different topologies are possible based on above three modes
 - ▶ Peer-to-Peer topology
 - ▶ Client/Server topology

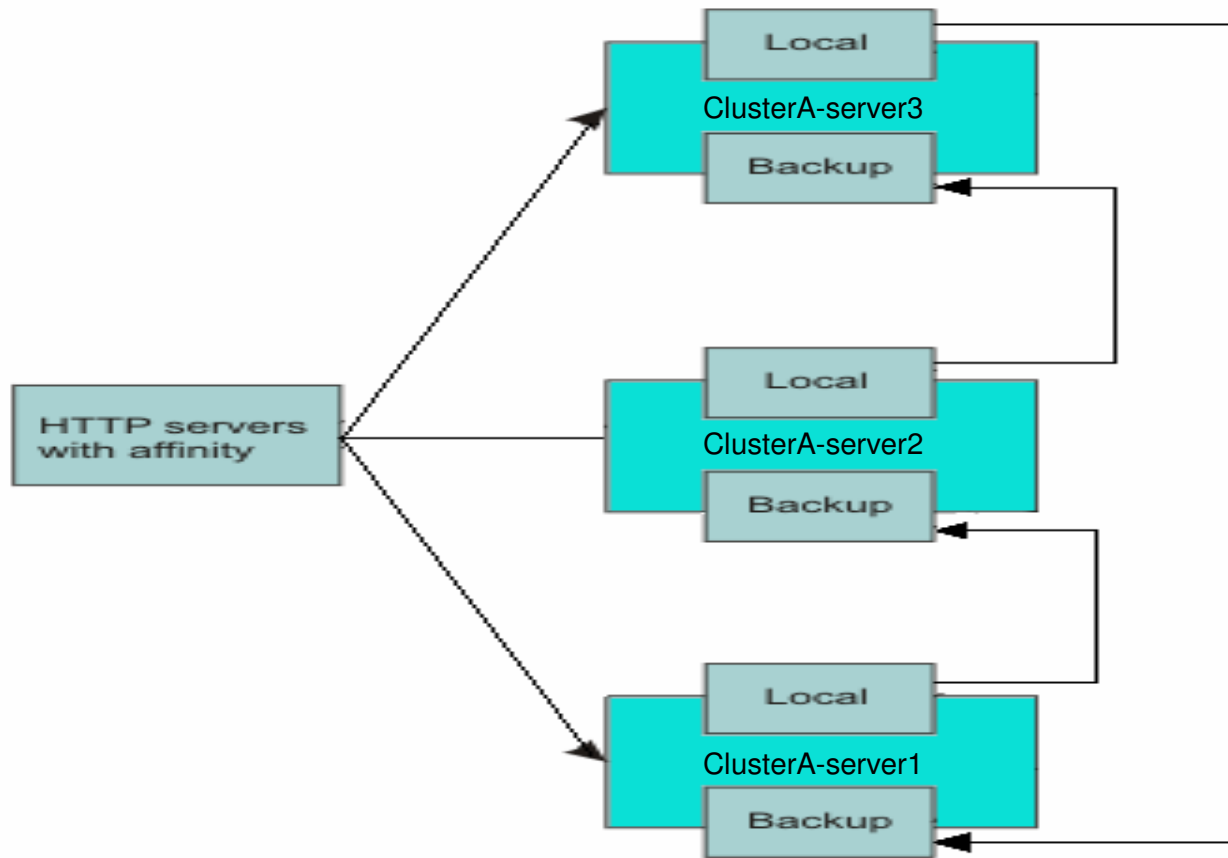
Advantages of Memory-to-Memory Replication

- Flexible configuration options, such as peer-to-peer and client/server
- Elimination of the overhead and cost of setting up and maintaining a real-time production database
- Elimination of single point of failure that can occur with a database
- Encrypted session information between application servers

Peer-to-Peer topology

- Each server stores a session in its own memory and also stores sessions to, and retrieves from, other application servers.
- Each application server acts as **client** by retrieving session from other servers and acts as **server** by providing sessions to other application servers
- This is the default configuration and each application server can:
 - ▶ Host the Web Application leveraging the HTTP Session
 - ▶ Send Changes to the HTTP Session that it owns
 - ▶ Receive backup copies of the HTTP session from all of the other servers in the cluster
- Most consolidated topology where the various system parts are co-located and requires the fewest server processes
- Most Stable implementation is achieved when each node has equal capabilities (CPU, memory and so on) and each handles the same amount of work

Peer-to-Peer replication topology



Hot Failover

- New feature since V6
- Plug-in knows where the backup session data is and performs failover to the server where backup session data is present
- Eliminates additional overhead of session retrieval when serving requests.
- Session manager encodes partition Id in JSESSIONID and the plug-in uses dynamic partition table from application server rather than using static clone information
 - ▶ Typical JSESSIONID : 0001VYXW0EKQQ3WDAGP43OX1Z0I:uk742k92.CacheID + SessionID + CloneID
 - ▶ With HotFailover functionality .. JSESSIONID : CacheID + SessionID + PartitionID
 - ▶ Partition table at plugin from application server via \$WSP http header
<PID_clone1>;<CID_clone1>:<PID_clone2>;<CID_clone2>:<PID_clone3>;<CID_clone3>
 - ▶ After server1 fails
<PID_clone1>;<CID_clone2>:<PID_clone2>;<CID_clone2>:<PID_clone3>;<CID_clone3>
- This feature is available only for peer-to-peer topology
- Plug-in must be at 6.0.2 or higher

Configuring peer-to-peer replication

- Create Replication domain
 - ▶ Configure request timeout
 - ▶ Encryption type
 - ▶ Number of replica
- Go to first cluster member by navigating to Servers > Application servers > *First Cluster member* > Session management > Distributed environment settings > Memory-to-memory replication
- For Replication domain, select the one just created
- Select “Both Client and Server” as replication mode
- Click Ok and Save to master configuration
- Repeat above steps for all the cluster members

Example Topology

Cluster Topology

This page displays the list of application server clusters in a tree format.

Local Topology

- Cell
 - ClusterB
 - ClusterA
 - Nodes
 - ashish1Node01 (ND 6.1.0.15)
 - Cluster members
 - ClusterA-server1
 - ClusterA-server2
 - ClusterA-server3

While creating cluster

Admin console > Servers > Clusters > New

Create a new cluster

Create a new cluster

→ **Step 1: Enter basic cluster information**

Step 2: Create first cluster member

Step 3: Create additional cluster members

Step 4: Summary

Enter basic cluster information

* Cluster name
ClusterA

Prefer local. Specifies whether enterprise bean requests will be routed to the node on which the client resides when possible.

Configure HTTP session memory-to-memory replication

Next Cancel

Create replication domain

Admin console > Environment > Replication domains > new

The screenshot shows the 'Replication domains > New' configuration page in the IBM Admin console. The page title is 'Replication domains' and the breadcrumb is 'Replication domains > New'. The main heading is 'Replication domains > New' and the instruction is 'Use this page to configure the replication properties that all of the components of this replication domain use.' The page is divided into sections: 'Configuration', 'General Properties', 'Encryption', and 'Number of replicas'. The 'General Properties' section includes a required field for 'Name' with the value 'SessionReplica' and a required field for 'Request timeout' with the value '5' seconds. The 'Encryption' section includes a dropdown for 'Encryption type' set to 'none' and a 'Regenerate encryption key' button. The 'Number of replicas' section includes radio buttons for 'Single replica' (selected), 'Entire Domain', and 'Specify', with a 'Number of replicas' input field below the 'Specify' option. At the bottom, there are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'.

Replication domains ?

Replication domains > New

Use this page to configure the replication properties that all of the components of this replication domain use.

Configuration

General Properties

* Name
SessionReplica

* Request timeout
5 seconds

Encryption

Encryption type
none

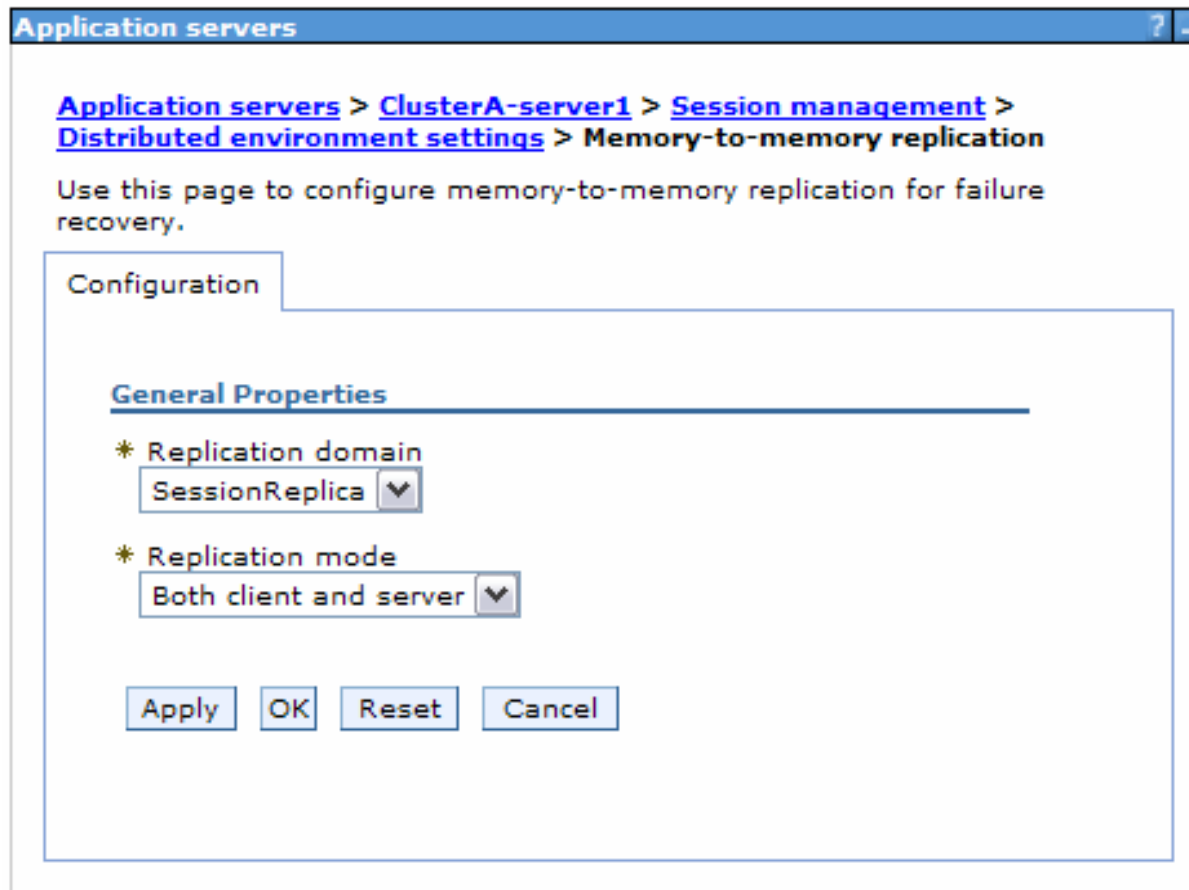
Regenerate encryption key

Number of replicas

Single replica
 Entire Domain
 Specify
Number of replicas

Apply OK Reset Cancel

Configure cluster member for peer-to-peer



The screenshot shows a web browser window titled "Application servers". The breadcrumb navigation is: [Application servers](#) > [ClusterA-server1](#) > [Session management](#) > [Distributed environment settings](#) > [Memory-to-memory replication](#). Below the breadcrumb is a descriptive text: "Use this page to configure memory-to-memory replication for failure recovery." There is a "Configuration" tab selected. Underneath, there is a section titled "General Properties" with two required fields: "Replication domain" set to "SessionReplica" and "Replication mode" set to "Both client and server". At the bottom of the configuration area are four buttons: "Apply", "OK", "Reset", and "Cancel".

Application servers

[Application servers](#) > [ClusterA-server1](#) > [Session management](#) > [Distributed environment settings](#) > [Memory-to-memory replication](#)

Use this page to configure memory-to-memory replication for failure recovery.

Configuration

General Properties

* Replication domain
SessionReplica ▼

* Replication mode
Both client and server ▼

Apply OK Reset Cancel





Restart Cluster to pickup the changes



Server clusters

Use this page to change the configuration settings for a cluster. A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

⊕ Preferences

New Delete Start Stop Ripplestart ImmediateStop

Select	Name	Status
<input checked="" type="checkbox"/>	ClusterA	
<input type="checkbox"/>	ClusterB	

Total 2

Advantages and Disadvantages peer-to peer

Advantages

- No additional processes and products are required to avoid a single point of failure.
- Reduce the time and cost required to configure and maintain additional processes or products

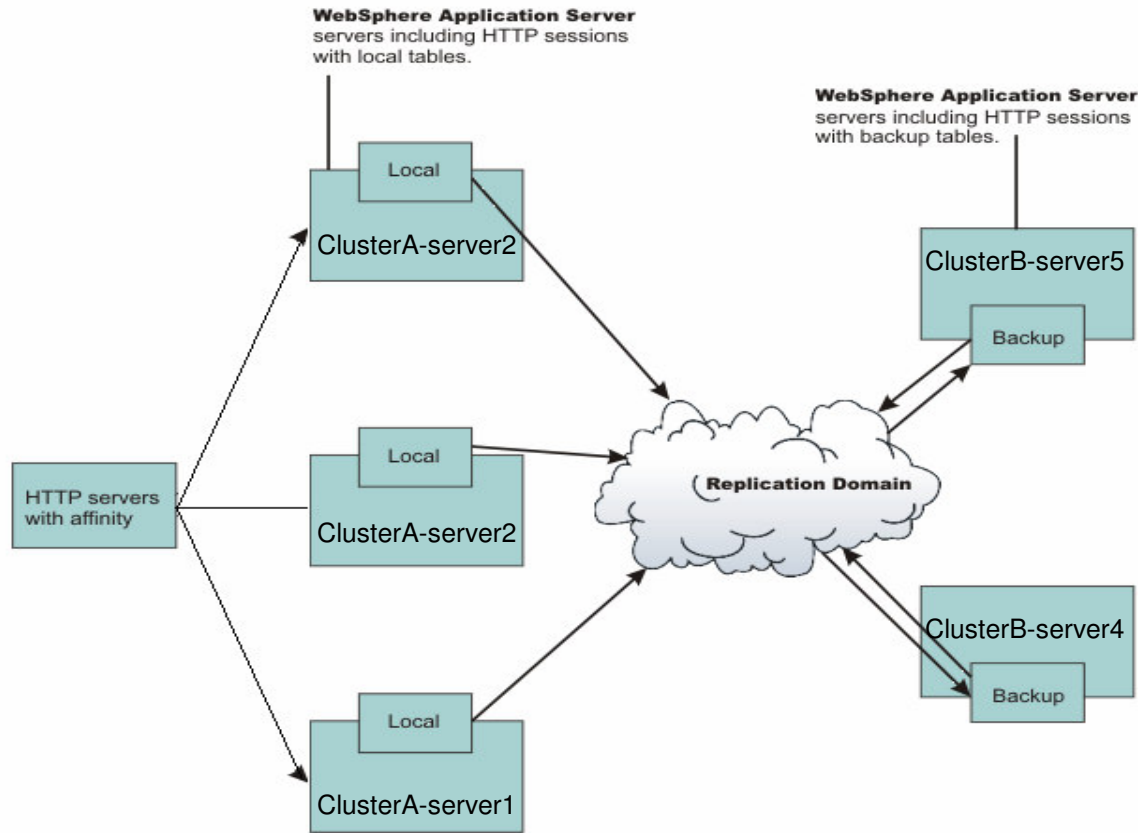
Disadvantages

- It can consume large amounts of memory with many users. This can cause a performance impact
- Does not provide stable configuration for cluster with only 2 members

Client/Server topology

- Application Servers act as either a replication Client or a Server
- Replication servers store sessions in their own memory and provide session info to clients.
- Dedicated replication servers just store session but do not respond to users' requests
- Client application servers send session information to replication servers and retrieve as needed. Clients respond to user requests
- Application servers configured as Server can not host any applications. They act as pure data store for backup sessions and receive updates from Clients
- Backup data is isolated into a separate JVM
- Client replicas will contain only local session data
- Application servers configured as Servers must be started before starting Clients

Client/Server topology



Example Topology














Cluster Topology ? -

Cluster Topology

This page displays the list of application server clusters in a tree format.

Local Topology

Cell

- [-]  [ClusterB](#)
 - [-]  Nodes
 - [-]  [ashish1Node01 \(ND 6.1.0.15\)](#)
 - [-]  Cluster members
 -  [ClusterB-server4](#)
 -  [ClusterB-server5](#)
- [-]  [ClusterA](#)
 - [-]  Nodes
 - [-]  [ashish1Node01 \(ND 6.1.0.15\)](#)
 - [-]  Cluster members
 -  [ClusterA-server1](#)
 -  [ClusterA-server2](#)
 -  [ClusterA-server3](#)

Create replication domain

Admin console > Environment > Replication domains > new

The screenshot shows a web-based configuration window titled "Replication domains" with a sub-header "Replication domains > New". Below the header is a brief instruction: "Use this page to configure the replication properties that all of the components of this replication domain use." The main content area is divided into sections: "Configuration" (which is the active tab), "General Properties", "Encryption", and "Number of replicas".

General Properties

- * Name:
- * Request timeout: seconds

Encryption

Encryption type: (dropdown menu)

Number of replicas

- Single replica
- Entire Domain
- Specify

Number of replicas:

At the bottom of the form are four buttons: , , , and .

Configure ClusterA members as Client

Application servers ? -

[Application servers](#) > [ClusterA-server1](#) > [Session management](#) > [Distributed environment settings](#) > [Memory-to-memory replication](#)

Use this page to configure memory-to-memory replication for failure recovery.

Configuration

General Properties

- * Replication domain
SessionReplica ▼
- * Replication mode
Client only ▼

Apply OK Reset Cancel

Configure ClusterB members as Server

Application servers

[Application servers](#) > [ClusterB-server4](#) > [Session management](#) > [Distributed environment settings](#) > **Memory-to-memory replication**

Use this page to configure memory-to-memory replication for failure recovery.

Configuration

General Properties

* Replication domain
SessionReplica ▼

* Replication mode
Server only ▼

Apply OK Reset Cancel





Start Server cluster ClusterB first



Server clusters

Use this page to change the configuration settings for a cluster. A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

+ Preferences

New Delete Start Stop Ripplestart ImmediateStop

Select	Name	Status
<input type="checkbox"/>	ClusterA	
<input type="checkbox"/>	ClusterB	

Total 2





Start Client cluster ClusterA second



Server clusters

Use this page to change the configuration settings for a cluster. A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

⊕ Preferences

New Delete Start Stop Ripplestart ImmediateStop

Select	Name	Status
<input type="checkbox"/>	ClusterA	
<input type="checkbox"/>	ClusterB	

Total 2

Advantages and Disadvantages of client/server

Advantages

- Clearly distinguishes the role of client and server.
- Reduces the consumption of memory on each application server
- Better performance
- Allows recycle of a backup server without affecting the servers running applications
- Better choice if your hardware varies across the cluster

Disadvantages

- Additional application servers have to be configured and maintained over and above those that interact with the user
- Need multiple replication servers to avoid a single point of failure

Best Practices

- All session managers connected to same replication domain should use same replication topology
- Create a separate replication domain for each consumer. i.e. one for session manager and other for dynamic cache
- Session manager and stateful session bean failover can be configured on same replication domain
- Multiple data replication service instances on same application server due to session manager memory-to-memory configuration at various levels that are configured to be part of the same domain must have the same mode
- Using multiple replication instances at various levels on same application server is not recommended

Tuning tips:

Application servers

[Application servers](#) > [ClusterA-server1](#) > [Session management](#) > [Distributed environment settings](#) > [Tuning parameters](#)

Use this page to select the session manager tuning options for managing session data in a distributed environment. These tuning options apply to the Web container only.

Configuration

General Properties

Tuning level:

- Very high (optimize for performance)
 - Write frequency Time based: 300 seconds
 - Write contents Only updated attributes
 - Schedule sessions cleanup: true
- High
 - Write frequency Time based: 300 seconds
 - Write contents All session attributes
 - Schedule sessions cleanup: false
- Medium
 - Write frequency End of servlet service
 - Write contents Only updated attributes
 - Schedule sessions cleanup: false
- Low (optimize for failover)
 - Write frequency End of servlet service
 - Write contents All session attributes
 - Schedule sessions cleanup: false
- [Custom settings](#)

Apply OK Reset Cancel



Tuning tips:

Application servers ?

[Application servers](#) > [ClusterA-server1](#) > [Session management](#) > [Distributed environment settings](#) > [Tuning parameters](#) > [Custom tuning parameters](#)

Use this page to specify tuning parameters for session management.

Configuration

General Properties

Write frequency

End of servlet service

Manual update

Time based:
 seconds

Write contents

Only updated attributes

All session attributes

Schedule sessions cleanup:

Specifies distributed sessions cleanup schedule

First time of day (0-23):

Second time of day (0-23):



Things to consider

Maximum in-memory session count:
 sessions

Allow overflow

Session timeout:

No timeout

Set timeout

minutes

- Once memory-to-memory replication is enabled, “Allow overflow” will be always true
- When overflow occurs, overflowed sessions will reside in the backup server
- If backup server goes down, all overflowed sessions will be lost
- Invalidation thread runs on backup server once memory to memory replication is enabled.
- Peer-to-peer topology with two cluster members, if one goes down, invalidation does not occur as expected

Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:
<http://www.ibm.com/developerworks/websphere/community/>
- Learn about other upcoming webcasts, conferences and events:
http://www.ibm.com/software/websphere/events_1.html
- Join the Global WebSphere User Group Community: <http://www.websphere.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:
<http://www.ibm.com/software/info/education/assistant>
- View a Flash replay with step-by-step instructions for using the Electronic Service Request (ESR) tool for submitting problems electronically:
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My support emails:
<http://www.ibm.com/software/support/einfo.html>

Questions and Answers

