**jbase\_agent and action\_timeout**

jbase\_agent and action\_timeout

An error "Action Timeout occurred". The error will occur if the jbase agent is killed after the timeout value specified in actionTimeout configuration property defined in JCA connection factory level is reached.  To know the timeout settings, please look up the custom properties details for the corresponding connection factory.  
  
**Resource adapters > T24ResourceAdapter > J2C connection factories > T24 JCA Connection Factory > Custom properties**  
  
To explain further, if the request is taking more than the actionTimeout value will be terminated and the corresponding connection will get moved to the connection pool. The objective of having the actionTimeout is to kill the long-running threads and make the connections available for the requests incoming.  
  
**ActionTimeout/ConnectionTimeout setting:**  
  
actionTimeout -> The T24 request will be processed by a subroutine call from OFSMessageMDB. This parameter configures the length of time the jbase\_agent will wait for a request to return a result before forcibly stopping the subroutine call and exiting. A message retry will be performed when the JMS message is passed to OFSMessageMDB again and a new jbase\_agent connection is created. This setting is very important to avoid T24 deadlocks as all T24 locks will be released when the action is stopped, thus allowing processing in other connections to continue. A good starting value would be 30 seconds.  
  
Generally, the browser’s connection timeout should be higher than the actionTimeout value. ConnectionTimeout(BrowserWeb.war/browserParameters.xml) > actionTimeout value.  
  
For example, connectionTimeout – 200 secs & actionTimeout – 180 secs  
  
**Note:** Set based on your business requirement.  
  
In order to avoid the reported error, kindly check with branch users and make sure all the enquiries/reports extraction/transactions are responded without any delay. If it requires optimization then optimize the enquiry by giving selection criteria or by indexing the respective fields. Just make sure all enquiries respond faster in order to make the free connections. So that connections will be available for the incoming new requests. If require any further assistance in fine-tuning the enquiry, contact your local T24 team.

**How to keep the jbase agent session idle without getting shutdown?**

[Vinay Jain - Admin](https://basecamp.temenos.com/s/profile/0056A000000izetQAA)  •  Jun 15, 2021  •  Technology > Core Infra > System Core (EB)

**From R09 to R21**

      49

Kindly be informed that the connection pools and killing/refreshing of the connections will be controlled and maintained by the Jboss Application server. There will not be any involvement in maintaining the connection pools by our T24 components such as tocfT24ra-ra.rar and tocfee.ear.  
  
Kindly go through the below link and set up the below and let us know whether it helps.  
  
<https://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/beta500/html/ch13s04s03s01.html>  
  
**Example 1:**  
  
<min-pool-size>5</min-pool-size>  
<max-pool-size>5</max-pool-size>  
<prefill>true</prefill>  
<idle-timeout-minutes>15</idle-timeout-minutes>  
  
With the above setup, if all the 5 sessions idle for more than 15 minutes then all will get killed and newly 5 jbase agents will be launched. Jboss will maintain 5 minimum sessions for processing the requests.  
  
**Example 2:**  
  
<min-pool-size>5</min-pool-size>  
<max-pool-size>5</max-pool-size>  
<prefill>false<prefill>  
<idle-timeout-minutes>15</idle-timeout-minutes>  
  
With the above setup, if all the 5 sessions idle for more than 15 minutes then all will get killed. Jboss will not maintain 5 minimum sessions for processing the requests and whenever the request comes, it will launch the new jbase agent threads for serving the requests.  
  
**Example 3:**  
  
<min-pool-size>5</min-pool-size>  
<max-pool-size>5</max-pool-size>  
<prefill>false</prefill>  
<idle-timeout-minutes>0</idle-timeout-minutes>  
  
With the above setup, the jbase agent sessions will not get killed. 0 - unlimited interval.  
  
Kindly verify the above and set accordingly as per your requirement. For more details about the parameters, kindly coordinate with your Jboss Administrator as the launching of the JCA connections will be controlled by Jboss Application Server.

Keywords - Jboss and connection and pool,Jboss and min-pool-size,Jboss and idle-timeout-minutes,Jboss and prefill,

**Pooling parameters**

* *<no-tx-separate-pools>* - whether separate subpools should be created for connections inside and outside JTA transactions (default false).
* *<min-pool-size>* - the minimum number of connections in the pool (default 0 - zero)
* *<max-pool-size>* - the maximum number of connections in the pool (default 20)
* *<blocking-timeout-millis>* - the length of time to wait for a connection to become available when all the connections are checked out (default 5000 == 5 seconds, from 3.2.4 it is 30000 == 30 seconds)
* *<idle-timeout-minutes>* - the number of minutes after which unused connections are closed (default 15 minutes)
* *<track-connection-by-tx>* - whether the connection should be *"locked"* to the transaction, returning it to the pool at the end of the transaction; in pre-JBoss-5.x releases the default value for Local connection factories is true and false for XA; since JBoss-5.x the default value is true for both Local and XA and the element is deprecated.
* *<interleaving/>* - enables interleaving for XA connection factories (this feature was added in JBoss-5.x)
* *<prefill>* - whether to attempt to prefill the connection pool to the minimum number of connections. NOTE: only supporting pools (OnePool) support this feature. A warning can be found in the logs if the pool does not support this. This feature is available in JBoss 4.0.5 and above.
* *<background-validation>* - In JBoss 4.0.5, background connection validation was added to reduce the overall load on the RDBMS system when validating a connection. When using this feature, JBoss will attempt to validate the current connections in the pool as a seperate thread (ConnectionValidator).
* *<background-validation-minutes>* - The interval, in minutes, that the ConnectionValidator will run. NOTE: It is prudent to set this value to something greater or less than the *<idle-timeout-minutes>*
* *<use-fast-fail>* - Whether or not to continue to attempt to acquire a connection from the pool even if the nth attempt has failed. False by default. This is to address performance issues where SQL validation may take significant time and resources to execute.

**How to stop jbase\_agent?**

[Vinay Jain - Admin](https://basecamp.temenos.com/s/profile/0056A000000izetQAA)  •  Feb 22, 2021  •  Technology > Core Infra > System Core (EB)

**From Prior to R05 to R06**

      19

How to stop jbase\_agent?

"jbase\_agent stop" command can be used if and only if jbase\_agent is started in jBASE background mode using the below command:

**jbase\_agent -p <listenPort> start (or) jbase\_agent start**

On executing the above command, a file**[ named jagent.lock<listenPort>]** will be created in /tmp directory. Therefore when jbase\_agent is stopped using the command **"jbase\_agent stop"** the system will look for a file named **jagent.lock20002** to obtain the pid of the jbase\_agent process running in background mode. Once the jbase\_agent process is shutdown the file **jagent.lock20002** [or jagent.lock<listenPort>] will be removed from /tmp directory.

Note: If you are running jbase\_agent in a different port say 20003 then you have executed "jbase\_agent -p 20003 stop" to stop the corresponding jbase\_agent process

However, if jbase\_agent is started using the below command:  
**nohup jbase\_agent -p 13331 &**  
the jbase\_agent process will run in Unix background mode and therefore no file will be created in /tmp directory

**Too many child processes spawned by jbase\_agent**

[Vinay Jain - Admin](https://basecamp.temenos.com/s/profile/0056A000000izetQAA)  •  Apr 07, 2021  •  Technology > Core Infra > System Core (EB)

      43

Too many child processes spawned by jbase\_agent

Kindly note, the number of jbase\_agent will be spawned based on the min/max pool setting done at the configuration properties of the JCA connection factory.   
    
  If the min session has been set as '1' only one jbase agent will be started at jbase startup. Based on the request flow to T24, it will increase up to max session and remain the same until system becomes idle. Based on the parameters like the <idle-timeout-minutes> jbase agent sessions can be killed after the timeout setting.  
    
Below are the main parameters to control jbase agents spawning in the below application servers.  
 In Jboss 6.4 EAP,   
   
 <min-pool-size>5</min-pool-size>  - the minimum number of connections maintained in the pool  
<max-pool-size>5</max-pool-size>   - the maximum number of connections allowed in the pool.  
<prefill>true</prefill>  
<idle-timeout-minutes>15</idle-timeout-minutes>  
  
Unless <prefill> is true then the pool will remain empty until first use at which point the pool will be filled to the <min-pool-size>.  When the pool size drops below the <min-pool-size> due to idle timeouts the pool will be refilled to the <min-pool-size>. Default is 0.  
<idle-timeout-minutes> - indicates the maximum time a connection may be idle before being closed. Setting to 0 disables it.  The default is 15 minutes.  
Kindly refer to the below link,

https://docs.jboss.org/jbossas/docs/Server\_Configuration\_Guide/beta500/html/ch13s04s03s01.html

Example 1:

<min-pool-size>5</min-pool-size>

<max-pool-size>5</max-pool-size>

<prefill>true</prefill>

<idle-timeout-minutes>15</idle-timeout-minutes>

With the above setup, if all the 5 sessions idle for more than 15 minutes then all will get killed and newly 5 jbase agents will be launched. Jboss will maintain 5 minimum sessions for processing the requests.

Example 2:

<min-pool-size>5</min-pool-size>

<max-pool-size>5</max-pool-size>

<prefill>false</prefill>

<idle-timeout-minutes>15</idle-timeout-minutes>

With the above setup, if all the 5 sessions idle for more than 15 minutes then all will get killed. Jboss will not maintain 5 minimum sessions for processing the requests and whenever the request comes, it will launch the new jbase agent threads for serving the requests.

Example 3:

<min-pool-size>5</min-pool-size>

<max-pool-size>5</max-pool-size>

<prefill>false</prefill>

<idle-timeout-minutes>0</idle-timeout-minutes>

With the above setup, the jbase agent sessions will not get killed. 0 - unlimited interval.

Kindly verify the above and set accordingly as per your requirement.