

## JRun Closed Connection

Posted At : June 18, 2004 7:59 AM | Posted By : Steven Erat

Related Categories: Events, ColdFusion, Macromedia

If you've noticed an increase in the frequency of "JRun Closed Connection" errors since having migrated to ColdFusion MX 6.1, then you may want to know one factor that is very likely contributing to that increase.

Speaking generally, web requests from the client browser are received by the web server and passed to the JRun webserver connector stub. The webserver connector stub communicates with the JRun server on the JNDI port over TCP/IP, typically port 51010. The web request is sent from the connector to the JRun server, and the JRun server will either assign the request to a JVM thread in the running request pool for immediate execution or JRun will assign the request to a thread in the queued request pool if the running pool is full.

The time that a thread is permitted to actively run is governed by the general timeout value in the ColdFusion Administrator on the Settings page, or it can be governed at the page level by the CFSETTING tag attribute 'requesttimeout'. Note that while a running thread is actively waiting on a third party resource such as when waiting for a database result set after having submitted the sql statement, the thread will not timeout. However, if in this example the database does return the query result after the timeout period has expired, the ColdFusion server will throw a timeout exception at that point. Finding error messages in the logs for CFQUERY timeouts is misleading because the thread already waited for some indefinite and lengthy period of time and finally received the result, but a timeout exception wouldn't have occurred until after that query completed. If the db result never comes back the thread will remain active in the running request pool indefinitely.

Should a request be assigned to a thread in the queued pool, the thread will wait for an opportunity to run up to some maximum time limit. If the request remains in the queued pool for that maximum period it will be timed out according to the setting **threadWaitTimeout** in jrun.xml. The timeout will cause JRun server to end the connection to connector for that queued thread, and the connector will report back to the webserver with the error **JRun Closed Connection**. The browser would show the following unhandled error:

### Server Error

The server encountered an internal error and was unable to complete your request.

JRun closed connection.

The other side of the coin is in the default-error.log for that server where the JRun server reports an error based on its perspective:

## Request timed out waiting for an available thread to run

In CFMX 6.0 to CFMX 6.0 Updater 3, the `threadWaitTimeout` value was set to 300 (seconds). This means that a queued request would wait 5 minutes for before timing out if it didn't get a chance to run. In CFMX 6.1, this value was changed to just 20 seconds as a default, and so an application that frequently maintained a large queue while experiencing low throughput would start reporting the connection error with a much higher frequency.

There are several approaches to resolve this.

- The Simultaneous Requests value could be increased
- The "slow" pages could be reviewed to determine when, how often, and why they run so long.
- The queue timeout could be increased

To increase the queue wait timeout:

1. Find `{cfusionmx}/runtime/servers/default/SERVER-INF/jrun.xml`
2. Find the xml node in that file for `"jrun.servlet.jrpp.JRunProxyService"`
3. In that node, find 20
4. Edit the value 20 to the desired value in seconds (previously 300)
5. Restart the CFMX server

While there are other ways to get the 'JRun Closed Connection' error, the unique characteristic to identify the shortened queue timeout problem is to match it up with 'Request timed out waiting for an available thread to run' errors in `default-error.log`. If you can correlate both errors together, then this is the problem you're having.

Related Links:

- [Search this blog for information on thread dumps](#)
- [Understanding web server connectors](#)
- [Technote for 'Request timed out waiting'](#)

#### Related Blogs:

- [JRun Threadpool settings](#), by Rupesh Kumar, ColdFusion Development