



# **WebSocket-based Edge MicroServer Release Notes**

**Version 5.4.0  
November 2017**

**Copyright © 2017 PTC Inc. and/or Its Subsidiary Companies. All Rights Reserved.**

User and training guides and related documentation from PTC Inc. and its subsidiary companies (collectively “PTC”) are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes. Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION. PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

**Important Copyright, Trademark, Patent, and Licensing Information:** See the About Box, or copyright notice, of your PTC software.

**UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND**

PTC software products and software documentation are “commercial items” as that term is defined at 48 C.F.R. 2.101. Pursuant to Federal Acquisition Regulation (FAR) 12.212 (a)-(b) (Computer Software) (MAY 2014) for civilian agencies or the Defense Federal Acquisition Regulation Supplement (DFARS) at 227.7202-1 (a) (Policy) and 227.7202-3 (a) (Rights in commercial computer software or commercial computer software documentation) (FEB 2014) for the Department of Defense, PTC software products and software documentation are provided to the U.S. Government under the PTC commercial license agreement. Use, duplication or disclosure by the U.S. Government is subject solely to the terms and conditions set forth in the applicable PTC software license agreement.

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA

# ThingWorx WebSocket-based Edge MicroServer (WS EMS) Release Notes

---

The new features and the bug fixes that have been made for the various releases of the ThingWorx WebSocket-based Edge MicroServer (WS EMS) are listed in the sections below. Starting with release 5.3.1, the IDs and Salesforce IDs for any issues that are fixed in this release are in a separate column (**ID / SFID**). The version(s) of the C SDK used by the version of WS EMS appears in parentheses in the table title.

To download the latest distribution bundle for your platform, visit the Software Downloads page of the PTC eSupport Portal, [https://support.ptc.com/appserver/cs/software\\_update/swupdate.jsp](https://support.ptc.com/appserver/cs/software_update/swupdate.jsp).

## WS EMS Version 5.4.0 (C SDK 2.0.4)

The 5.4.0 release of the WS EMS is built on release 2.0.4 of the ThingWorx C SDK. See the C SDK release notes for information about release 2.0.0 through 2.0.4. The following table lists the enhancements and issues fixed in this release.

ID (SFID)	Description
<b>Enhancements</b>	
EDGA-1135	<p>Print warnings to the log when insecure configuration is used (LSR/EMS).</p> <p>Insecure HTTP Server configurations will now cause the WS EMS and LSR to log warning messages to the log when any one or more of the following conditions is true:</p> <ul style="list-style-type: none"><li>• SSL is disabled. (The <code>http_server.ssl</code> property is set to <code>false</code>.)</li><li>• Authentication is disabled.</li><li>• Certificate validation is disabled.</li><li>• Self-signed certificates are allowed.</li></ul>
EDGA-1085	<p>Make <code>config.json.complete</code> contain valid JSON.</p> <p>The <code>config.json.complete</code> file is now a valid JSON file that can be loaded and parsed by the WS EMS. The values in this file are the same default values as in <code>config.json</code>. See also EDGA-1084.</p>
EDGA-1084	<p>Rename <code>config.json.complete</code> to <code>config.json.documented</code>.</p>

ID (SFID)	Description
	<p>The original <code>config.json.complete</code> has been renamed to <code>config.json.documented</code> to serve as a reference when configuring the WS EMS. It is important to note that <code>config.json.documented</code> is NOT a valid JSON file for use with the WS EMS. If you want to use all of the configuration options, use <code>config.json.complete</code>. See EDGA-1085</p>
EDGA-1071	<p>Expose HTTP Server <code>max_clients</code> value to the <code>config.json</code> (i.e., configuration files) of the WS EMS.</p> <p>Previously, the HTTP Server of the WS EMS was hard-coded to allow only a maximum of 16 concurrent clients to be handled at a single time. The LSR defaults to a maximum of 16, but allows the user to override this value by setting the <code>scripts.max_clients</code> value in <code>config.lua</code>. For this release, the <code>max_clients</code> property has been added to the <code>http_server</code> group in all of the configuration files for the WS EMS. <code>max_clients</code> denotes the maximum number of HTTP clients that can be served concurrently by the WS EMS.</p> <p>In addition, the <code>ports_to_try</code> property has been added to the <code>http_server</code> group in all of the configuration files for the WS EMS, providing complete control over the HTTP Server.</p>
EDGA-1065	<p>Use UTC Timestamps in the WS EMS log.</p> <p>The logger of the WS EMS now uses UTC timestamps instead of local time when writing to a log file.</p>
EDGA-1039	<p>Print out version number of WS EMS on startup.</p> <p>After WS EMS has been initialized, it displays its version or release number on the console and writes the number to the log file as an INFO level log message. For example, with this release, WS EMS would print out 5.4.0.</p>
EDGA-1038	<p>Make the FIPS switch functional at runtime for the WS EMS.</p> <p>The existing <code>#ifdef</code> for the FIPS switch has been removed. A configuration option for enabling FIPS mode has been added to the <code>config.json</code>, <code>config.json.complete</code>, and <code>config.json.documented</code> configuration files for the WS EMS. By default, FIPS mode is disabled. The WS EMS will check if FIPS mode is enabled on startup.</p>
EDGA-1028	<p>Create Windows WS EMS build based on latest OpenSSL libraries.</p>

ID (SFID)	Description
	<p>This release of the WS EMS provides version 1.0.2L of the OpenSSL libraries for the supported Windows platform. In addition, the WS EMS distribution bundles for Windows now contain either the OpenSSL libraries or the axTLS library. The distribution bundle that contains OpenSSL libraries has <code>openssl</code> in its file name. Similarly, the bundle that contain axTLS has <code>axtls</code> in its file name. Choose a distribution bundle based on the operating system, SSL/TLS libraries you want to use, and platform. All distribution bundles that contain the OpenSSL libraries also support FIPS mode. By default FIPS mode is disabled. If you want to use FIPS mode, you need to change the configuration.</p>
EDGA-1023	<p>Create Linux builds based on OpenSSL for the WS EMS.</p> <p>As of this release, the WS EMS provides binaries for the latest version of OpenSSL, 1.0.2L . These binaries are provided for Linux 32-bit, Linux 64-bit, Linux ARM, and Linux ARM-HWFPU platforms. In addition, the WS EMS distribution bundles for Linux now contain either OpenSSL libraries or the axTLS library. The distribution bundles that contain OpenSSL libraries have <code>openssl</code> in their file names. Similarly, the bundles that contain axTLS have <code>axtls</code> in their file names. Choose a distribution bundle based on the operating system, SSL/TLS libraries you want to use, and platform. All distribution bundles that contain the OpenSSL libraries also support FIPS mode. By default FIPS mode is disabled. If you want to use FIPS mode, you need to change the configuration.</p>
EDGA-923	<p>Load PEM-encoded private key/certificate from disk.</p> <p>The WS EMS now supports the following use cases:</p> <ul style="list-style-type: none"> <li>• Loading a PEM-encoded certificate from disk</li> <li>• Loading a PEM-encoded private key from disk with a passphrase</li> </ul>
EDGA-922	<p>Regenerate axTLS configuration to remove default key/certificate.</p> <p>The axTLS configuration has been changed to allow the use of a custom private key/certificate. It is strongly recommended, however, that you use the OpenSSL 1.0.2L library that is provided in the distribution bundles of the WS EMS.</p>
EDGA-641	<p>Add support for System D to the WS EMS installation scripts.</p>


ID (SFID)	Description
	Previously the <code>install.sh</code> script for the WS EMS did not support Linux distributions that use System D. The <code>install.sh</code> script now supports System D. See also EDGA-640.
EDGA-610	<p>Logging output configuration for LuaScriptResource.</p> <p>The Lua Script Resource and WS EMS use the same logging library (<code>libLogger</code>). The WS EMS had many more configuration options for the logger exposed in <code>config.json</code> files than the LSR did in <code>config.lua</code>. The LSR now has the same logging output configuration options as the WS EMS.</p>
<b>Issues Fixed in This Release</b>	
EDGA-1150	<p>Update EMS documentation to use correct REST URL in examples.</p> <p>The example REST URLs now all use Thingworx instead of ThingWorx.</p>
EDGA-982 / 13648635	<p><code>config.json.complete</code> in the <code>etc</code> directory is not a valid JSON file.</p> <p>The enhancements provided by EDGA-1085 and EDGA-1084 resolve this issue.</p>
EDGA-981 / 13648635	<p>Syntax errors in <code>config.json.complete</code> file and WS EMS help center and guide.</p> <p>RESOLUTION: The <code>config.json.complete</code> file now has the missing commas. The documentation has been updated to match this configuration file. See also EDGA-1085 for additional changes for <code>config.json.complete</code> and the addition of a new, fully commented file, which should NOT be used to run WS EMS.</p>
EDGA-640 / 13325589	<p>WS EMS failed to install on Ubuntu 16.04 due to <code>systemd</code> Init System.</p> <p>The <code>install.sh</code> script for WS EMS now supports <code>systemD</code>.</p>

ID (SFID)	Description
<b>Known Issue</b>	
EDGA-1105	<p>Refactor EMS 'testBoundThing' service to stop SSL_READ errors on LSR.</p> <p>The WS EMS has a service that runs periodically to test if a bound thing on a remote host still exists. If TLS is enabled, this service test can result in read errors on the remote host, since the WS EMS will open and close the socket but not send any data. There read errors, such as those seen in the LSR logs below, are benign and can be ignored.</p> <pre>[DEBUG] 2017-07-11 17:06:48,943 SDK:       twTlsServer_Accept: Client Handshake in progress [ERROR] 2017-07-11 17:06:48,948 SDK:       TW_SSL_READ: Error reading from SSL stream [ERROR] 2017-07-11 17:06:48,948 SDK: TW_SSL_READ:       Timed out or error waiting reading       from socket.       Error: error:00000000:lib(0):func(0):reason(0) [DEBUG] 2017-07-11 21:06:48,956 TlsStream::doclose:       Disconeccting socket</pre> <p>The TW_SSL_READ is calling <code>SSL_read()</code>, which will return 0. This return value indicates that the remote peer may have just shut down the connection.</p>

### WS EMS Version 5.3.4 (C SDK 1.5.1 and 1.5.2)

WS EMS v.5.3.4 is built on C SDK v.1.5.2, which includes changes for C SDK v.1.5.1. See the C SDK release notes for information about those releases.

ID / SFID	Description
<b>Enhancements</b>	
EDGA-1035	<p>Limitations on log files have been added to the logging persistence function. The total log size on disk will not exceed the configured value. A new property, <code>buffer_size</code>, allows you to specify the maximum size of a single log message (in bytes).</p> <p>In addition, the property, <code>flush_chunk_size</code>, has been added to allow you to the number of bytes to write before flushing to disk.</p> <p>These properties are available in the <code>config.json.complete</code> configuration file in WS EMS installation.</p>
EDGA-1034	The same format is now used in log messages written to the

ID / SFID	Description
	console as in log messages written to the persisted log files. The log messages are no longer wrapped in a JSON object. The persisted log files are just text files. Their content will match what is printed out on the console.
EDGA-909	The timestamps for log messages on WS EMS now show the actual time rather than the time that the log messages were written to the stream in the logger thread. This change applies to both WS EMS and the Lua Script Resource (LSR).
<b>Issue Fixed in This Release</b>	
EDGA-1050 / 13318364	Asset Deployments Failing, Requiring LSR Restart When an asset deployment fails as a result of WS EMS disconnecting and reconnecting during a download, any subsequent deployments to that asset fail until the Lua Script Resource (LSR) is restarted.
<b>Special Note</b>	
 <b>Caution</b> As of release 8.1 of ThingWorx Core, PTC is ending the life of the ThingWorx XMPP Edge MicroServer. The XMPP EMS is no longer available to any new customer. New customers should use the ThingWorx WebSocket-based Edge MicroServer (WS EMS) instead.	


### WS EMS Version 5.3.3 (C SDK 1.5.0, which includes C SDK 1.4.0 & 1.4.1)

ID / SFID	Description
<b>Enhancement</b>	
EDGA-811	Add new configuration option to <code>config.json</code> ( <code>tick_resolution</code> ).  The documentation for WS EMS has been updated for the change in the C SDK (CSDK-862) that has been merged into WS EMS for this release.
<b>Issues Fixed in This Release</b>	
EDGA-829 13609759	Stopping the WS EMS overwrites any changes in the <code>config.json</code> file.  This issue is fixed in this release.
EDGA_818 13603198	The new FIPS EMS v5.3.2.1693 crashes when first connecting, even if the specified port is wrong.  This issue is fixed in this release by the merge with C SDK 1.4.1.
EDGA-735 13318364	LSR can hit 50% CPU when waiting for file transfers to finish  Package deployments in ThingWorx Utilities SCM failed because the file did not fully transfer to the edge device. When this happens, the LSR hit the CPU at 50%.  This issue is fixed in this release.






## WS EMS Version 5.3.2.1693 — Issues Fixed (C SDK 1.3.5)

ID / SFID	Description
EDGA-567	<p>Attempting to run the <code>install.bat</code> file in order to run WS EMS and Lua Script Resource as services was failing on the <code>sc create</code> lines. Instead of creating a service, the help message for the command was displayed.</p> <p>This issue is fixed in this release.</p>
EDGA-546 / C12947309	<p>WS EMS could not reconnect to ThingWorx server via a proxy server.</p> <p>This issue was fixed by changes in the C SDK 1.3.5.</p>
EDGA-141	<p>When all the parameters of <code>config.json</code> were not contained within curly brackets (<code>{}</code>), the WS EMS would report an error, overwrite the existing <code>config.json</code> file, and exit.</p> <p>This issue is fixed in this release.</p>
EDGA-78	<p>The WS EMS distribution bundle now includes <code>doc</code> directory that contains the PDF of the <i>ThingWorx WebSocket-based Edge MicroServer (WS EMS) Developer's Guide</i> for this release and a <code>/doc</code> directory that contains the luadoc files. In addition, the <code>doc/index.htm</code> file has been removed.</p>
<b>Known Issue</b>	
EDGE-1964 / CSDK-14	<p>The Edge device (WS EMS) cannot establish a secure websocket connection (WSS, SSL) to a ThingWorx server. The error appears as <code>Error 0, Error initializing SSL connection, twWs_Connect: Error restarting socket. Error 0, and/or No compatible ciphers</code> when a WS EMS device attempts to connect. This issue applies to the C SDK 1.3.2 through 1.3.5; the .NET SDK 5.6.2, through 5.6.4, the WS EMS 5.3.2.x, and the iOS SDK 1.0.</p> <p>CAUSE: Versions of Apache Tomcat 8.0.35 and above have disabled RSA-based ciphers by default due to forward secrecy concerns. (see <a href="https://tomcat.apache.org/tomcat-8.0-doc/changelog.html">https://tomcat.apache.org/tomcat-8.0-doc/changelog.html</a> for 8.0.34.). The axTLS libraries used by the WS EMS (and all ThingWorx C SDK, .NET SDK, and iOS SDK) support two encryption ciphers: <code>TLS_RSA_WITH_AES_256_CBC_SHA</code> and <code>TLS_RSA_WITH_AES_128_CBC_SHA</code>. Any application that uses SSL for Edge connections if the Tomcat server is upgraded to 8.0.35 or later may be affected by this change to Tomcat.</p> <p>WORKAROUNDS:</p> <ul style="list-style-type: none"> <li>Downgrade to a version of Tomcat version 8.0.33 or lower.</li> <li>In the <code>server.xml</code> configuration file of Tomcat, explicitly define a list of ciphers that includes the axTLS ciphers. For an example with a list of ciphers supported in Tomcat version 8.0.36, see <a href="https://support.ptc.com/appserver/cs/view/solution.jsp?n=CS245522">https://support.ptc.com/appserver/cs/view/solution.jsp?n=CS245522</a>.</li> </ul> <p> <b>Caution</b></p> <p>Make sure that any ciphers you use have been validated with any internal security requirements before implementing this workaround in production environment</p>
This release includes fixes that were made for the C SDK, versions 1.3.3, 1.3.4, and 1.3.5.	

## WS EMS Version 5.3.2 — Enhancements and Issues Fixed (C SDK 1.3.2)

ID	Description
<b>Enhancements</b>	
EDGE-975	UpdateSubscribedPropertyValues is now always triggered after property updates.
EDGE-239	A section on using FIPS has been added to the user guide for WS EMS. This documentation also includes information from the fix for EDGE-1250 (enabling client authentication).
EDGA-80	This release includes a subdirectory, doc, that contains the *.luadoc files that provide details for the LuaScriptResource.
EDGA-78	<p>The EMS distributions now include the following items:</p> <ul style="list-style-type: none"> <li>A doc directory that contains the user's guide.</li> <li>A subdirectory, doc/luadoc, that contains the microserver/doc/*.luadoc files.</li> </ul> <p>In addition, the distributions no longer contain a file called version.txt.</p>
EDGA-72	connect_retries is missing from config.json.complete. This property has been added to config.json.complete for this release.
<b>Issues Fixed</b>	
EDGE-1485	The FIPS build of the WS EMS for this release enables you to set up a secure connection to the ThingWorx Core on Windows 7 machines.
EDGE-1250	Client authentication cannot be enabled for the C SDK. The C SDK uses the axTLS library for authentication. axTLS does NOT support client authentication. However, the WS EMS provides a build that contains OpenSSL and FIPS (select the bundle that has "FIPS" in its name). Use this build when client authentication and FIPS mode are required. See also the new section on FIPS in the PDF that accompanies the WS EMS bundle.
EDGE-1076	<p>The following timeouts are now documented in config.json.complete and can be read from config.json by the WS EMS:</p> <ul style="list-style-type: none"> <li>socket_read_timeout</li> <li>frame_read_timeout</li> <li>ssl_read_timeout</li> </ul>
EDGE-874	<p>The WS EMS was responding very slowly to requests, in comparison to v.5.2.2 and 5.3.0.</p> <p>This issue is fixed in this release.</p>
EDGE-758	<p>When calling the GetRemoteMetadata service from the ThingWorx Core via a Connection Server (v.6.5.11, 7.0, and 7.0.1), the Connection Server logs an error and the ThingWorx Core service times out. The WS EMS is successfully receiving the request and sending packets back to the Connection Server.</p> <p>This issue is fixed in this release.</p>
EDGA-346	<p>Memory leak while decoding JSON into InfoTable under certain conditions.</p> <p>This issue is fixed in this release.</p>
EDGA-345	<p>Investigate memory leaks in EMS</p> <p>This issue is fixed in this release.</p>

ID	Description
EDGA-344	PUT request caused memory leak in EMS.  This issue is fixed in this release.
EDGA-226	FIPS EMS Crashes.  This issue is fixed in this release.
EDGA-217	EMS Memory Leaks and Crashes on Linux.  This issue is fixed in this release.
EDGA-211	The <code>wsems -version</code> command now returns the correct version.
EDGA-178	The WS EMS now passes proxy configuration settings to the Tunnel Manager so that in a network that is set up to route all traffic through a proxy, the tunnel requests are no longer blocked.
EDGA-123	When you start WS EMS without a <code>config.json</code> file, you now will see error messages explaining what has happened. The WS EMS will try to load an existing <code>.booted</code> configuration file when the <code>config.json</code> is missing. When it fails to find a <code>.booted</code> file, it goes back to the original. When that fails again, the WS EMS will tell you that it failed to load any configuration file. In addition, if the <code>config.json</code> file is not formatted correctly, the WS EMS will report an appropriate error message.
<b>Known Issue</b>	
EDGE-1964 / CSDK-14	<p>The Edge device (WS EMS) cannot establish a secure Websocket connection (WSS, SSL) to a ThingWorx server. The error appears as <code>Error 0, Error initializing SSL connection, twWs_Connect: Error restarting socket. Error 0, and/or No compatible ciphers</code> when a WS EMS device attempts to connect. This issue applies the C SDK 1.3.2, .NET SDK 5.6.2 and 5.6.3, WS EMS 5.3.2, and iOS SDK 1.1.</p> <p>CAUSE: Versions of Apache Tomcat 8.0.35 and above have disabled RSA-based ciphers by default due to forward secrecy concerns. (see <a href="https://tomcat.apache.org/tomcat-8.0-doc/changelog.html">https://tomcat.apache.org/tomcat-8.0-doc/changelog.html</a> for 8.0.34.). The axTLS libraries used by the WS EMS (and all ThingWorx C SDK, .NET SDK, and iOS SDK) support two encryption ciphers: <code>TLS_RSA_WITH_AES_256_CBC_SHA</code> and <code>TLS_RSA_WITH_AES_128_CBC_SHA</code>. Any application that uses SSL for Edge connections if the Tomcat server is upgraded to 8.0.35 or later may be affected by this change to Tomcat.</p> <p>WORKAROUNDS:</p> <ul style="list-style-type: none"> <li>• Downgrade to a version of Tomcat version 8.0.33 or lower.</li> <li>• In the <code>server.xml</code> configuration file of Tomcat, explicitly define a list of ciphers that includes the axTLS ciphers. For an example with a list of ciphers supported in Tomcat version 8.0.36, see <a href="https://support.ptc.com/appserver/cs/view/solution.jsp?n=CS245522">https://support.ptc.com/appserver/cs/view/solution.jsp?n=CS245522</a>.</li> </ul> <p> <b>Caution</b></p> <p>Make sure that any ciphers you use have been validated with any internal security requirements before implementing this workaround in production environment</p>
This release also includes fixes/improvements that were made for the C SDK 1.3.2.	

## WS EMS Version 5.3.1 — Enhancements and Issues Fixed (C SDK 1.3.1)

ID	Description
<b>Enhancements</b>	
EDGE-953	The <code>tw_dir.pwd()</code> function has been added to the Lua Script Resource for this release.
EDGE-890	The <code>config.json.complete</code> file has been updated to reflect recent changes. In particular, you can no longer specify an array of ThingWorx Core addresses for the connection from WS EMS to the ThingWorx Core. You can only specify ONE destination host and port. If you have Microservers that have this configuration, note that this version of WS EMS does not error when it encounters the array. It tries the first address and, if that fails, it returns an error to that effect.
EDGE-831	Add inputs to install scripts for the EMS that allow renaming of the services.  The inputs already existed in the Windows install script and are now documented. The inputs for Linux scripts have been added and are documented in the <i>ThingWorx WebSocket-based Edge MicroServer Developer's Guide</i> (PDF) that accompanies the WS EMS distribution.
EDGE-821	The <i>ThingWorx WebSocket-based Edge MicroServer Developer's Guide</i> (PDF) now provides the versions of the libraries required for use on supported Linux platforms.
EDGE-837	The API documentation (luadoc) for Lua has been added to the WS EMS distribution bundle.
EDGE-706	The <i>ThingWorx WebSocket-based Edge MicroServer Developer's Guide</i> (PDF) has been extensively revised for this release. In addition, it now documents the REST API supported by the WS EMS.
EDGE-363	The install scripts for the WS EMS on Linux have been enhanced to support other platforms.
<b>Issues Fixed</b>	
EDGE-829	The ListFiles service gives different result for WS EMS 5.3 and 5.0.  The ListFiles service in this release (5.3.1) now returns the path without the file name, as it did in release 5.0.
EDGE-823/EDGE-499 Case 12819599	The WS EMS running as a service on a Netbiter ec350 device failed to start up and displayed the following message: <code>Error creating BSD socket</code> .  This release resolves this issue.
EDGE-818	EMS crashes (SIGABRT) during LSR startup on some Linux platforms.  This problem occurs only on Linux systems with <code>libc.so.6-2.6</code> or older, which are not supported. Refer to the revised <i>ThingWorx WebSocket-based Edge MicroServer Developer's Guide</i> that accompanies this release for information about the C libraries that are required. The new section with this information is in Chapter 2 and is called "Libraries".
EDGE-803	Cannot POST events through EMS REST interface.  This release resolves this issue.
EDGE-762	Updating multiple properties using REST API call via WS EMS error. This issue has been resolved. See Tech Support Article 000225416.
EDGE-756	WS EMS does not connect with offline storage.

ID	Description
	This release resolves this issue.
EDGE-752	The PUTJson service strips the URI <code>query</code> parameter. This release resolves this issue.
EDGE-680	Offline storage stores data when turned off in the configuration file. This release resolves this issue.
EDGE-605	WS EMS cannot save the <code>config.json.booted</code> file when the <code>-cfg</code> flag is used. This release resolves this issue.
This release also includes fixes/improvements that were made for the C SDK 1.3.1.	

## WS EMS Version 5.3.0 (C SDK 1.3.0)

New Features and Fixes
<ul style="list-style-type: none"> <li>• The Content Loader services have been modified. In earlier releases, services were too strict when they interpreted the content-type of response headers.</li> <li>• WS EMS now handles requests made by the Content Loader services for any bound thing.</li> <li>• The script resource no longer prepends the <code>*</code> character to the <code>p_data</code> file of an Identifier.</li> <li>• Duplicate entries in <code>GetDirectoryStructure</code> have been removed.</li> <li>• Various memory leaks have been fixed.</li> <li>• The distribution bundle of this release includes an updated version of the document, <i>WebSocket Edge MicroServer (WS EMS) User's Guide</i>.</li> <li>• This release also includes fixes/improvements that were made for the C SDK 1.3.0.</li> </ul>

## WS EMS Version 5.2.2 (C SDK 1.3.0)

### New Features

- This release contains an updated Lua script to facilitate the functionality that updates software, which is part of the ThingWorx Converge RSM application.

## WS EMS Version 5.2.0 (C SDK 1.3.0)

### New Features

- The WS EMS now uses the C SDK for its WebSocket library.
- This release also includes fixes/improvements that were made for the C SDK 1.3.0.

### Bug Fixes

- The HTTP server now uses the SDK `twSocket`, even in non-SSL mode.
- For the HTTP server, you can now configure the timeout setting for reading content.
- The issue with `AxTlsStream` in the Linux version of the HTTP server is fixed.
- The script, `modbus.lua`, has been updated with fixes from the Technical Sales department.
- A deadlock that was caused by the request to unbind in certain situations has been fixed.
- The bug in the WS EMS handler that removes resources has been fixed.
- LSR (Lua Script Resource) scripts can now exit out of a `tw_utils.psleep()` call when a script is shut down.
- The staging directory of WS EMS can now reference a virtual directory (`virtual_dir`) or a directory on the file system.
- Fix for EDGE-256: The LSR now includes the correct information about data shapes when browsing the properties in an infotable.
- Fix for EDGE-186: An asterisk (\*) is no longer prepended to the Identifier; the EMS now connects on second startup.
- The default size of the buffer of `TlsStream` has been changed to 16K.
- The console is now more responsive.
- An issue wherein the EMS would shut down while it tried to connect has been fixed.
- Fix for EDGE-303: The software update now works in Lua with 5.0.
- Support for OpenSSL FIPS support has been added for Win32 platforms.
- A bug that caused large multipart messages to fail has been fixed.

## WS EMS Version 5.1.0.8

<b>New Features</b>
<ul style="list-style-type: none"><li>• The WS EMS now supports transfers of files whose name or path contain multi-byte characters. This feature includes virtual directories that are configured at the server.</li></ul>
<b>Bug Fixes</b>
<ul style="list-style-type: none"><li>• The handling of incoming messages that occurs within the <code>sendMessageBlocking</code> function has been fixed so that the function handles responses only. This fix avoids deadlocks in certain situations.</li><li>• The <code>twMessage_Send</code> function has been changed to check to see if the EMS is authenticated before it sends.</li><li>• The code that sends offline messages to insert a new <code>RequestId</code> has been changed in order to remove any potential conflicts from a previous ID.</li><li>• Mutex protection has been added in the <code>twTlsClient_Reconnect</code> and <code>twTlsClient_ConnectSession</code> functions.</li><li>• A segment fault that occurs while the EMS stores non-persistent, offline messages has been fixed.</li><li>• The copyright for documentation has been updated.</li><li>• The Location property is now registered so that it shows up when browsed.</li></ul>

## WS EMS Version 5.0.4.121

<b>New Features</b>
<p>This release includes changes to the way that WS EMS validates SSL certificates. The default behavior has been changed so that WS EMS does NOT accept self-signed certificates, and always validates the SSL certificate provided by the ThingWorx server. This change can result in the following errors at startup:</p> <ul style="list-style-type: none"><li>• If you are currently connecting to a ThingWorx Core instance that uses a self-signed certificate, you must explicitly enable the acceptance of self-signed certificates in your WS EMS configuration.</li><li>• If you are currently connecting to a ThingWorx Core instance that uses a certificate that has been signed by a trusted certificate authority (CA), you must obtain the root certificate of that CA, in <code>.pem</code> format. You must then deploy that root certificate with your EMS. Alternatively, you can disable certificate validation (NOT recommended, especially in a production environment).</li></ul> <p>You can change the following configuration options in the 'certificates' section of in your <code>config.json</code> file:</p> <pre>"certificates" : {   "validate": true   false, // Enable/disable certificate validation   "cert_chain": [ "/path/to/ca_root.pem" ], // Inform EMS about CA root cert   "allow_self_signed": true   false // Accept self signed cert from server }</pre> <p><b>Note on signed certificates:</b></p> <p>Certificates in the certificate chain of the server must be signed, using one of the following signing algorithms: SHA1, MD 5, or MD2.</p>