WebSocket-based Edge MicroServer Release Notes

Version 5.3.4
June 2017

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 WebSocke-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.

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

ID / SFID	Description
Enhancements	
EDGA-1035	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, `buffer_size`, allows you to specify the maximum size of a single log message (in bytes). In addition, the property, `flush_chunk_size`, has been added to allow you to specify the number of bytes to write before flushing to disk. These properties are available in the `config.json.complete` configuration file in the WS EMS installation.
EDGA-1034	The same format is now used in log messages written to the 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 the 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).

WS EMS Version 5.3.4 (C SDK 1.5.1) (continued)

ID / SFID	Description
Issue Fixed in This Release	
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.

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

ID / SFID	Description
Enhancement	
EDGA-811	Add new configuration option to `config.json` (`tick_resolution`). The documentation for WS EMS has been updated for the change in the C SDK (CSDK-862) that has been merged into the WS EMS for this release.
Issues Fixed in This Release	
EDGA-829 13609759	Stopping the EMS overwrites any changes in the `config.json` 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.
EDGA-682 13414038	LSR pushes null property (with Value 0) to platform when property retrieval fails Properties are initialized with the value 0. When the `getProperties` service and subsequent handler read call are made, they return a 500 error response. However, the start script does not check the response and just sets the property as if the value has changed. The LSR is setting properties to 0, as they are initialized with that value, every time they cannot be retrieved. Error handling has been added to the code that verifies the response type from `getProperties`. The change has been added to `template.lua`.
EDGA-648 13394597, 13420582	luaScriptResources stop working with error message "`bad argument #1 to 'pairs' (table expected, not nil)`" This issue is fixed in this release.
EDGA-600	Default values for `auto_bind` `host` and `port` not being used when running in non-gateway mode. the Windows 7 example for WS EMS 5.3.2 failed with the error `emsRequestHandler: Thing not bound to EMS or host is not set.`

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

ID / SFID	Description

	The example is fixed in this release. The code that works is:
	`"auto_bind" [`
	`"localhost","ptcRemoteThing",false}`
	`]`
	The host and port must be specified when using `auto_bind`.

EDGA-581	Remove MODBUS scripts from EMS distribution. The `etc/thingworx/scripts`, `etc/thingworx/lua` and `etc/custom/templates` directories of the WS EMS distribution no longer include MODBUS files and pre-compiled Lua binaries. The documentation for the WS EMS has been updated with these changes for this release.
EDGA-560 13271857	Tunnel `max_concurrent` setting does NOT limit concurrent VNC access to EMS. The following settings are not supported by WS EMS: `max_concurrent`, `buffer_size`, `read_timeout`, and `idle_timeout` are not supported by WS EMS. The WS EMS documentation has been updated to remove `buffer_size`, `read_timeout`, and `idle_timeout`.
EDGA-227	Specifying an incorrect path in the virtual directory configuration of EMS can cause a core dump This issue is fixed in this release.
EDGA-218	EMS not reporting duty cycle. With the duty cycle set to 50%, the WS EMS connects to ThingWorx server properly, but the generated Thing object does not show up as disconnected or disappear at any time. There are no log messages to indicate that the WS EMS went offline. This issue is fixed in this release.

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

ID	Description
EDGA-567	Attempting to run the `install.bat` file in order to run WS EMS and Lua Script Resource as services was failing on the `sc create` lines. Instead of creating a service, the help message for the command was displayed. This issue is fixed in this release.
EDGA-141	When all the parameters of `config.json` were not contained within curly brackets ({}), the WS EMS would report an error, overwrite the existing `config.json` file, and exit. This issue is fixed in this release.
EDGA-78	The WS EMS distribution bundle now includes `doc` directory that contains the PDF of the *ThingWorx WebSocket-based Edge MicroServer (WS EMS) Developer's Guide* for this release and a `/doc` directory that contains the luadoc files. In addition, the `doc/index.htm` file has been removed.

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

ID	Description
Known Issue	
EDGE-1964 / CSDK-14	The Edge device (WS EMS) cannot establish a secure Websocket connection (WSS, SSL) to a ThingWorx server. The error appears as `Error 0`, `Error initializing SSL connection`, `twWs_Connect: Error restarting socket. Error 0`, and/or `No compatible ciphers` 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. CAUSE: Versions of Apache Tomcat 8.0.35 and above have disabled RSA-based ciphers by default due to forward secrecy concerns. (see https://tomcat.apache.org/tomcat-8.0-doc/changelog.html 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: TLS_RSA_WITH_AES_256_CBC_SHA and TLS_RSA_WITH_AES_128_CBC_SHA. 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. WORKAROUNDS: • Downgrade to a version of Tomcat version 8.0.33 or lower. • In the `server.xml` 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 https://support.ptc.com/appserver/cs/view/solution.jsp?n=CS245522. **Caution** Make sure that any ciphers you use have been validated with any internal security requirements before implementing this workaround in production environment

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

ID	Description
Enhancements	
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	The EMS distributions now include the following items: • A `doc` directory that contains the user's guide. • A subdirectory, `doc/lua`, that contains the `microserver/doc/*.luadoc` files. In addition, the distributions no longer contain a file called `version.txt`.

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

ID	Description
EDGA-72	`connect_retries` is missing from `config.json.complete`. This property has been added to `config.json.complete` for this release.
Issues Fixed	
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	The following timeouts are now documented in config.json.complete and can be read from config.json by the WS EMS: • `socket_read_timeout` • `frame_read_timeout` • `ssl_read_timeout`
EDGE-874	The WS EMS was responding very slowly to requests, in comparison to v.5.2.2 and 5.3.0. This issue is fixed in this release.
EDGE-758	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. This issue is fixed in this release.
EDGA-346	Memory leak while decoding JSON into InfoTable under certain conditions. This issue is fixed in this release.
EDGA-345	Investigate memory leaks in EMS This issue is fixed in this release.
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 `wsems -version` 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 `config.json` file, you now will see error messages explaining what has happened. The WS EMS will try to load an existing `.booted` configuration file when the `config.json` is missing. When it fails to find a `.booted` 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

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

ID	Description
	`config.json` file is not formatted correctly, the WS EMS will report an appropriate error message.
Known Issue	
EDGE-1964 / CSDK-14	The Edge device (WS EMS) cannot establish a secure Websocket connection (WSS, SSL) to a ThingWorx server. The error appears as `Error 0`, `Error initializing SSL connection`, `twWs_Connect: Error restarting socket. Error 0`, and/or `No compatible ciphers` 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. CAUSE: Versions of Apache Tomcat 8.0.35 and above have disabled RSA-based ciphers by default due to forward secrecy concerns. (see https://tomcat.apache.org/tomcat-8.0-doc/changelog.html 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: TLS_RSA_WITH_AES_256_CBC_SHA and TLS_RSA_WITH_AES_128_CBC_SHA. 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. WORKAROUNDS: • Downgrade to a version of Tomcat version 8.0.33 or lower. • In the `server.xml` 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 https://support.ptc.com/appserver/cs/view/solution.jsp?n=CS245522. **Caution** Make sure that any ciphers you use have been validated with any internal security requirements before implementing this workaround in production environment

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

ID	Description
Enhancements	
EDGE-953	The `tw_dir.pwd()` function has been added to the Lua Script Resource for this release.
EDGE-890	The `config.json.complete` 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.

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

ID	Description
	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 *ThingWorx WebSocket-based Edge MicroServer Developer's Guide* (PDF) that accompanies the WS EMS distribution.
EDGE-821	The *ThingWorx WebSocket-based Edge MicroServer Developer's Guide* (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 *ThingWorx WebSocket-based Edge MicroServer Developer's Guide* (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.
Issues Fixed	
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: `Error creating BSD socket`. 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 libc.so.6-2.6 or older, which are not supported. Refer to the revised *ThingWorx WebSocket-based Edge MicroServer Developer's Guide* 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. This release resolves this issue.
EDGE-752	The PUTJson service strips the URI `query` 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 `config.json.booted` file when the `-cfg` flag is used. This release resolves this issue.

WS EMS Version 5.3.0 (C SDK 1.3.0)

New Features and Fixes

- The Content Loader services have been modified. In earlier releases, services were too strict when they interpreted the content-type of response headers.
- WS EMS now handles requests made by the Content Loader services for any bound thing.
- The script resource no longer prepends the * character to the `p_data` file of an Identifier.
- Duplicate entries in `GetDirectoryStructure` have been removed.
- Various memory leaks have been fixed.
- The distribution bundle of this release includes an updated version of the document, *WebSocket Edge MicroServer (WS EMS) User's Guide*.

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 v.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. See the C SDK Release Notes for v.1.3.0 on page .

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

New Features

- 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.

Bug Fixes

- The handling of incoming messages that occurs within the `sendMessageBlocking` function has been fixed so that the function handles responses only. This fix avoids deadlocks in certain situations.
- The `twMessage_Send` function has been changed to check to see if the EMS is authenticated before it sends.
- The code that sends offline messages to insert a new `RequestId` has been changed in order to remove any potential conflicts from a previous ID.
- Mutex protection has been added in the `twTlsClient_Reconnect` and `twTlsClient_ConnectSession` functions.
- A segment fault that occurs while the EMS stores non-persistent, offline messages has been fixed.
- The copyright for documentation has been updated.
- The Location property is now registered so that it shows up when browsed.

WS EMS Version 5.0.4.121

New Features

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:

- 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.
- 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 `.pem` format. You must then deploy that root certificate with your EMS. Alternatively, you can disable certificate validation (NOT recommended, especially in a production environment).

You can change the following configuration options in the 'certificates' section of in your `config.json` file:

```
"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
}
```

Note on signed certificates:

Certificates in the certificate chain of the server must be signed, using one of the following signing algorithms: SHA1, MD 5, or MD2.