WebSocket-based Edge MicroServer Release Notes

Version 5.4.5
December 2018

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*.

WS EMS Version 5.4.5 (C SDK 2.2.0)

The 5.4.5 release of the WS EMS is built on release 2.2.0.of the ThingWorx Edge C SDK, which means that it includes all the changes made for releases 2.1.5 and 2.2.0 of the C SDK. See the C SDK release notes for information about the changes in the 2.2.0 and 2.1.5 releases. The following table describes the enhancements and the issues fixed in this release of the WS EMS:

ID (SFID)	Description
Enhancements	
EDGA-1478	A new configuration option has been added to `config.json`, `config.complete`, and `config.documented` to allow you to specify what cipher suites are used by the edge device when communicating with the ThingWorx platform. It supports the OpenSSL Cipher List format, as described here: *https://www.openssl.org/docs/man1.0.2/apps/ciphers.html* This option is only supported on WS EMS releases that use OpenSSL; AxTLS releases ignore this option.
EDGA-1765	A new configuration option, `ws_connection.compression`, allows you to enable or disable websocket compression. This option is available in the `config.json`, `config.complete`, and `config.documented` files of the WS EMS. Previously, websocket compression was always enabled.
EDGA-1848 (14231681)	Updates have been made to the `ws_connection` default settings in `config.json.complete` and `config.json.documented` to better match settings used by the ThingWorx Edge C SDK. For example, the `message_idle_time` is now set to a default value of 50 seconds.
Issues Fixed in This Release	
EDGA-1778 (14211095)	When running the WS EMS as a service, it was possible for the WS EMS to become stuck in a state where it would not shut down properly when a 'stop service' request was made, requiring it to be killed. This issue has been fixed in this release.
EDGA-1836	If the WS EMS was run as a service and was able to connect but not successfully authenticate with ThingWorx within 60 seconds, it would 'hang' and not try to reconnect. This issue is fixed in this release.
EDGA-1854	This issue has been resolved as part of PTC's continued investment in helping customers reduce risks associated with security threats.

ID (SFID)	Description
EDGA-1855 (PSPT-5919)	The default certificate and private key have been removed from the WS EMS. This means that you must configure the HTTP Server of the WS EMS to use your own certificate and private key when running with SSL/TLS.

WS EMS Version 5.4.4 (C SDK 2.1.4)

The 5.4.4 release of the WS EMS is built on release 2.1.4 of the ThingWorx Edge C SDK, which includes all the changes made for release 2.1.3 of the C SDK. See the C SDK release notes for information about the changes in the 2.1.3 and 2.1.4 releases. The following table describes the issue fixed in this release:

ID (SFID)	Description
EDGA-1784	This fix resolves an edge case that could occur during file transfers when websocket compression was in use and could cause the file transfer to fail. .

WS EMS Version 5.4.3 (C SDK 2.1.2)

The 5.4.3 release of the WS EMS is built on release 2.1.2 of the ThingWorx Edge C SDK. See the C SDK release notes for information about the changes in that release. The following table lists the enhancement provided in this release:

ID (SFID)	Description
Enhancement	
EDGA-1471	Rebuild Duty Cycle in the WS EMS With this release, the behavior of WS EMS for the duty cycle feature has changed to enable it to track file transfers and tunnels, as well as property and service requests from the ThingWorx platform. Duty cycle will not disconnect the WS EMS from the platform if any of the following conditions are true: • A message has been received from the platform during the last `delay_duty_cycle` time interval. • A message has been sent to the platform but no response has been received yet. • A file transfer is pending or in progress. • A remote session (tunnel) is in progress (open). Finally the WS EMS will not be disconnected from the ThingWorx platform immediately after starting up. Instead, the WS EMS will disconnect at the next Duty Cycle event after startup. For more details, see the section on configuring duty cycle modulation in the ThingWorx WebSocket-based Edge MicroServer Developer's Guide, v.5.4.3, or the *ThingWorx Edge SDKs and WebSocket-based Edge MicroServer (WS EMS) Help Center* .

WS EMS Version 5.4.2 (C SDK 2.1.2)

The 5.4.2 release of the WS EMS is built on release 2.1.2 of the ThingWorx Edge C SDK. See the C SDK release notes for information about the changes in that release. The following table lists the issues fixed in this release:

ID (SFID)	Description
EDGA-1613	Changes have been made to how the Lua Script Resource's `/script` and `/scriptcontrol` REST endpoints work out-of-the-box. By default, you will not be able to use these endpoints to dynamically create, update, delete, or restart

ID (SFID)	Description
	scripts using the REST API. Any requests to these services will result in a `405 - Method Not Allowed` error. This feature can be enabled by adding the line `scripts.script_resource_enable_rest_services = true` to your `config.lua`
EDGA-1598	Move LSR index page off of / and into `help.html` For this release, the current index page contents for the LSR help has been moved to a help page. You can reach the help page at `/help.html`, `/help`, or `/help/`. The new index page is completely blank.
EDGA-1594 (14175756)	Add note in EMS/LSR documentation that usernames should not contain a ":" (colon) character. For this release, a new topic has been added to the WS EMS guide and the Help Center that addresses this issue, in the context of using REST APIs with WS EMS and LSR. See the developer's guide included in your distribution bundle or the *ThingWorx Edge SDKs and WebSocket-based Edge MicroServer (WS EMS) Help Center*. The topic title is Running REST API Calls with Postman on WS EMS and LSR..
EDGA-1593	EMS TRACE logs Basic Auth header. EMS was adding the value of the Authorization header on TRACE level. With this release, all Authorization headers are obfuscated in the logs.
EDGA-1566 (14167223)	Add CSRF token support to the REST API in the WS EMS and LSR. This change now requires any requests from a client that can change state (such as POST, PUT or DELETE) include a CSRF token in the headers of their request. This token will be provided by the server and put into response header with the key `x-csrf-token`. The client must include this same header and token value with any request that can change state. The token will change periodically based on the `csrf_token_rotation_period` value set in `config.json/config.lua`. The default period is every 10 minutes. Neither the WS EMS nor the LSR require changes or configuration updates to support CSRF tokens. The tokens are enabled by default. Applications that use the REST interface of the WS EMS or LSR will need to be updated to include the CSRF token, or CSRF protection must be disabled (not recommended). You can disable CSRF protection by adding the line `enable_csrf_token = false` in the `http_server struct` of `config.json` (WS EMS) or `scripts.script_resource_enable_csrf_token = false`` in `config.lua` (LSR). CSRF protection is enabled *only* when authorization is enabled as well. If authorization is disabled, no token values will be used. PTC recommends always using TLS, enabling authorization, and encrypting sensitive credentials in configuration files.

WS EMS Version 5.4.1 (C SDK 2.1.2)

The 5.4.1 release of the WS EMS is built on release 2.1.2 of the ThingWorx Edge C SDK. See the C SDK release notes for information about the changes in that release. The following table lists the issues fixed in this release:

ID (SFID)	Description
Enhancements	
EDGA-1409	EMS REST API service `TestPort` doesn't work as described.

ID (SFID)	Description

	CAUSE: The `TestPort` service was expecting a full infotable representation rather than the simplified infotable representation.
	
	RESOLUTION: The `TestPort` service now supports simplified infotables. In addition support for additional optional parameters has been expanded. For example:
	
	`{`
	`  "host": "127.0.0.1",  // Required`
	`  "port": "80",  // Required`
	`  "useSSL":false,  // Optional`
	`  "useProxy": false  // Optional`
	`}`

EDGA-1406	Update the topic for the `Restart` service. For any edge-side restart requests to work correctly, the `Restart` REST service relies on a previously undocumented configuration parameter (`restart`) to be set in the *`config.json`* file of the WS EMS. Information about this configuration option has been added to the developer's guide for the WS EMS and to the *ThingWorx Edge SDKs and WebSocket-based Edge MicroServer (WS EMS) Help Center*. The topic titles are "Viewing All Configuration Options" and "Restart" (the REST API).
EDGA-1102	Update Developer's Guide to include all LSR security-related options. A new group of topics, called "Configuring the Lua Script Resource", has been added to developer's guide included in your distribution bundle and to the *ThingWorx Edge SDKs and WebSocket-based Edge MicroServer (WS EMS) Help Center* The security-related topics in this section are "Configuring the HTTP Server (SSL/TLS Certificate)" and "Configuring the Connection to the WS EMS".
Issues Fixed in This Release	
EDGA-1543 (14154298)	The `rap_password` does not work using AES. CAUSE: An entry in `config.lua`did not accept an encrypted password. All other entries did/do accept the encrypted password. This release resolves this issue.
EDGA-1473	Update note about kEDH ciphers in EMS developer's guide. RESOLUTION: The note about cipher suites in the WS EMS Developer's Guide has been changed for this release to say the following: If your application communicates with an instance of the ThingWorx platform that uses Java 1.7, the cipher suite list should include !kEDH (as shown below) to disable ephemeral Diffie-Hellman ciphers . Otherwise, ephemeral Diffie-Hellman (EDH) key exchange will fail, and your WS EMS will be unable to connect to the platform. `<CipherSuites>DEFAULT:!kEDH</CipherSuites>`
EDGA-1470 (C14006638)	"TW_VALIDATE_CERT: Certificate rejected" is output on Start Tunnel with a Remote Access Widget, even though ThingWorx successfully connected with EDGE MicroServer 5.4.0 This issue is fixed in this release.

ID (SFID)	Description
EDGA-1469	EMS doesn't set tunnel TLS configuration settings unless a proxy is used. CAUSE: The initialization code of the WS EMS configured TLS settings on the tunnel Manager only if a proxy was set. RESOLUTION: The proxy check has been removed so that the tunnel settings are configured, regardless of a proxy being set.
EDGA-1465 Ubuntu 16.04 SF ID 14067293 Ubuntu 12.04.5 SF ID 14129246	Lua Script Resource seems to hang on Linux operating systems when trying to communicate with the EMS without TLS. RESOLUTION: The WS EMS and LSR can now communicate with each other, and things running the LSR show up as bound in ThingWorx Composer when the HTTP server of the WS EMS is not using SSL (`ssl = false`) and the LSR is not using SSL to communicate with the WS EMS (`rap_ssl = false`).
EDGA-1454	WS EMS HTTP Server logs IP addresses in network byte order. CAUSE: The HTTP Server of the WS EMS was writing IP addresses to the log in reverse order because IP addresses were stored internally in network byte order rather than host byte order. This issue is fixed in this release.
EDGA-1414 (14043263)	Configuring SSL/TLS certificates validation is inconsistent in ThingWorx Edge Microserver 5.4. In previous release of the WS EMS the `cert_chain` parameter expected an array value. This changed in 5.4.0 and `cert_chain` now expects a string that points to a single file that contains all Certificate Authority (CA) certificates used for validation. This change was not reflected throughout the Developer's Guide. RESOLUTION: `cert_chain` is now defined correctly as a string (`cert_chain : "/path/to/ca_root.pem",`) throughout the Developer's Guide..
EDGA-1408	EMS returning '402' instead of '403' for Forbidden errors. This issue is fixed in this release.
EDGA-1105	EMS causing `SSL_READ` errors to appear in the logs of the LSR when SSL is used.. This issue is fixed in this release.

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 Edge 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
Enhancements	
EDGA-1135	Print warnings to the log when insecure configuration is used (LSR/EMS). 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:

ID (SFID)	Description
	• SSL is disabled. (The `http_server.ssl` property is set to `false`.) • Authentication is disabled. • Certificate validation is disabled. • Self-signed certificates are allowed.
EDGA-1085	Make `config.json.complete` contain valid JSON. The `config.json.complete` 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 `config.json`. See also EDGA-1084.
EDGA-1084	Rename `config.json.complete` to `config.json.documented`. The original `config.json.complete` has been renamed to `config.json.documented` to serve as a reference when configuring the WS EMS. It is important to note that `config.json.documented` is NOT a valid JSON file for use with the WS EMS. If you want to use all of the configuration options, use `config.json.complete`. See EDGA-1085
EDGA-1071	Expose HTTP Server `max_clients` value to the `config.json` files (i.e., configuration files) of the WS EMS. 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 `scripts.max_clients` value in `config.lua`. For this release, the `max_clients` property has been added to the `http_server` group in all of the configuration files for the WS EMS. `max_clients` denotes the maximum number of HTTP clients that can be served concurrently by the WS EMS. In addition, the `ports_to_try` property has been added to the `http_server`group in all of the configuration files for the WS EMS, providing complete control over the HTTP Server.
EDGA-1065	Use UTC Timestamps in the WS EMS log. The logger of the WS EMS now uses UTC timestamps instead of local time when writing to a log file.
EDGA-1039	Print out EMS version number on startup. 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`.
EDGA-1038	Make the FIPS switch functional at runtime for the WS EMS. The existing `#ifdef` for the FIPS switch has been removed. A configuration option for enabling FIPS mode has been added to the `config.json`, `config.json.complete`, and `config.json.documented` configuration files for the WS emS. By default, FIPS mode is disabled. The WS EmS will check if FIPS mode is enabled on startup.
EDGA-1028	Create Windows build based on latest OpenSSL libraries.

ID (SFID)	Description
	This release of the WS EMS provides version 1.0.2L of the OpenSSL libraries. In addition, the WS EMS will use OpenSSL by default instead of axTLS. If you want to use axTLS, you need to change the configuration.
EDGA-1027	Create Windows EMS build based on latest OpenSSL libraries. This release provides the OpenSSL libraries for version 1.0.2L. The WS EMS will use these libraries by default for security instead of the axTLS library (which is still available in the distribution bundle, just no longer the default).
EDGA-1023	Create Linux builds based on OpenSSL for the WS EMS. As of this release, the WS EMS provides binaries for the latest version of OpenSSL, 1.0.2L . Both FIPS and non-FIPS binaries are provided for Linux 32–bit, Linux 64–bit, Linux ARM, and Linux ARM-HWFPU platforms.
EDGA-923	Load PEM-encoded private key/certificate from disk. The WS EMS now supports the following use cases: • Loading a PEM-encoded certificate from disk • Loading a PEM-encoded private key from disk with a passphrase
EDGA-922	Regenerate axTLS configuration to remove default key/certificate. 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.
EDGA-641	Add support for System D to the WS EMS installation scripts. Previously the `install.sh` script for the WS EMS did not support Linux distributions that use System D. The `install.sh` script now supports System D. See also EDGA-640.
EDGA-610	Logging output configuration for LuaScriptResource. The Lua Script Resource and WS EMS use the same logging library (`libLogger`). The WS EMS had many more configuration options for the logger exposed in `config.json` files than the LSR did in `config.lua`. The LSR now has the same logging output configuration options as the WS EMS.
Issues Fixed in This Release	
EDGA-1150	Update EMS documentation to use correct REST URL in examples. The example REST URLs now all use `Thingworx` instead of `ThingWorx`.
EDGA–982 / 13648635	`config.json.complete` in the `etc` directory is not a valid JSON file. The enhancements provided by EDGA-1085 and EDGA-1084 resolve this issue.
EDGA–981 /. 13648635	Syntax errors in `config.json.complete` file and WS EMS help center and guide. RESOLUTION: The `config.json.complete` file now has the missing commas. The documentation has been updated to match this configuration file. See also EDGA-1085 for additional changes for `config.json.complete` and the addition of a new, fully commented file, which should NOT be used to run WS EMS.

ID (SFID)	Description
EDGA-640 / 13325589	WS EMS failed to install on Ubuntu 16.04 due to `systemd` Init System. The `install.sh` script for WS EMS now supports `systemD`.
Known Issue	

EDGA–1105	Refactor EMS 'testBoundThing' service to stop SSL_READ errors on LSR.
	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.
	`[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`
	The `TW_SSL_READ` is calling `SSL_read()`, which will return 0. This return value indicates that the remote peer may have just shut down the connection.

WS EMS Version 5.3.4 (C SDK 1.5.1 and 1.5.2)

The 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
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 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-1031	Enforce a Sleep inside the Software Update State Machine. Certain states do not have anything that enforces an idle timeout AND they can spin a tight loop that can consume the CPU at 100 percent.

ID / SFID	Description
	RESOLUTION: After analyzing each state as to whether a sleep needs to be added in the "main campaign execution loop", a sleep has been added to the START_DOWNLOAD state. While in this state, the addition of the sleep prevents the possibility of a state sitting in a tight loop and consuming all of the CPU. Note that sleeps already did and still do exist in the DOWNLOADING, WAIT_FOR_DOWNLOAD, WAIT_FOR_INSTALL states. A sleep is not needed for the ABORTED, FAILED, DOWNLOADED, INSTALLING, NOTIFIED, and CREATED states.
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).
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.
Special Note	
 **Caution:** As of release 8.1 of ThingWorx platform, 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
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 WS EMS for this release.
Issues Fixed in This Release	
EDGA-829	Stopping the WS EMS overwrites any changes in the `config.json` file.
13609759	This issue is fixed in this release.
EDGA_818	The new FIPS EMS v5.3.2.1693 crashes when first connecting, even if the specified port is wrong.
13603198	This issue is fixed in this release by the merge with C SDK 1.4.1.
EDGA-735	LSR can hit 50% CPU when waiting for file transfers to finish
13318364	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	LSR pushes null property (with Value 0) to platform when property retrieval fails
13414038	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

ID / SFID	Description
	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`. The example is fixed in this release. The code that works is: `"auto_bind" [ {"name": "TestRemoteThing", "host": "localhost", "port": 8001, "gateway": false} ]` The host and port must be specified when using `auto_bind`.
EDGA-581	Remove MODBUS scripts from WS 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 the WS 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 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 WS 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 / SFID	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-546 / C12947309	WS EMS could not reconnect to ThingWorx server via a proxy server.

ID / SFID	Description
	This issue was fixed by changes in the C SDK 1.3.5.
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.
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
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
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`.

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 platform 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 platform via a Connection Server (v.6.5.11, 7.0, and 7.0.1), the Connection Server logs an error and the ThingWorx platform 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

ID	Description
	that fails again, the WS EMS will tell you that it failed to load any configuration file. In addition, if the `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
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
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 platform addresses for the connection from WS EMS to the ThingWorx platform. 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 *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.

ID	Description
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-499Case 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.
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
• 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*.

New Features and Fixes
• This release also includes fixes/improvements that were made for the C SDK 1.3.0.

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

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.

New Features

- 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 platform 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 platform 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.