UWC release

Release Notes

7 July 2021

## Version History/Revision History

These are the main releases of Product Name:

|  |  |  |
| --- | --- | --- |
| Date | Revision | Description |
| July 2021 | 0.1 | Sprint 30 and 31 |

## Customer Support

For technical support, including answers to questions not addressed in this product, visit the technical support forum, FAQs, and other support information at: <insert URL link here> or http://www.intel.com/software/products/support/.

Please remember to register your product at https://registrationcenter.intel.com/ by providing your email address. Registration entitles you to free technical support, product updates and upgrades for the duration of the support term. It also helps Intel recognize you as a valued customer in the support forum.

**NOTE**: If your distributor provides technical support for this product, please contact them for support rather than Intel.

To provide feedback and suggestions, go to the Graphics community page <https://communities.intel.com/community/tech/graphics>

To submit an issue, go to Intel® Premier Support: (<https://employeeportal.intel.com/irj/portal/IntelPremierSupportUser>)

For more information on registering to Intel Premier Support, go to: <http://software.intel.com/en-us/articles/performance-tools-for-software-developers-intel-premier-support>

Contents:

[1 Introduction 3](#_Toc68692543)

[2 New in This Release 4](#_Toc68692544)

[3 Fixed Issues 10](#_Toc68692545)

[4 Known Issues / Limitations 11](#_Toc68692546)

[5 Related Documentation 12](#_Toc68692547)

[6 Where to Find the Release 13](#_Toc68692548)

[7 Best Known Configuration (optional) 14](#_Toc68692549)

[8 Hardware and Software Compatibility 15](#_Toc68692550)

# Introduction

The Intel® UWC is a reference design for a secured management platform that gives third party application developers an easy access to data from O&G well-site

Solution is targeted to address multiple pain areas O&G industry is facing in day-to-day operations. These pain areas are further restricting O&G industry to get benefitted from technology advancements resulting from cloud-based services and applications for business intelligence (BI), analytics, dashboards, etc. There is a need to provide a uniform mechanism to connect, monitor and control various devices in an O&G well-site adhering to real-time nature of the industry.

To learn more about this product, see:

* New features listed in the [New in this Release](#_What's_New) section below, or in the help.
* Reference documentation listed in the [Related Documentation](#_Related_Documentation) section below

# New in This Release

## All Features List – Sprint Wise

**Sprint 1**

* Data polling mechanism
* Integration with EISv2.0 pv
* Softdel's TCP/IP Modbus stack integration
* YML parser for Modbus application
* Configuration file handler
* JSON parser for Modbus application
* Dev mode (non secure) support
* Error handling
* Sprint 1 IP plan
* Sprint 1 Detailed Design Doc

**Sprint 2**

* Modbus TCP module/container (in Prod mode) with support for data polling
  + The messages are published on ZeroMQ on topics specified in docker-compose.yml file.
* Modbus TCP module integrated with EIS v2.1-Alpha-RC4 release
  + Modbus TCP runs as a separate container as per EIS framework.
* Support for ETCD based configuration YML files
  + Provisioning script is updated to accept multiple YML files as input and store it in ETCD
* MQTT
  + MQTT Export and MQTT are running as container as per EIS framework
* Draft version of design document
* IP Plan

**Sprint 3**

* Write request support for Modbus TCP and RTU from MQTT
* UWC is integrated with EIS v2.1-Alpha release
* Modbus RTU feature
  + Modbus-RTU-Master will be running as container as per EIS framework
* Dynamic change support using ETCD
* MQTT restricted to local host
* Installer scripts for deployment
* Instrumentation for messages in container logs
* User Guide – Draft version v1.0
* IP Plan

**Sprint 4**

* Word and Byte swap for polling operation in Modbus TCP feature
* Word and Byte swap for polling operation in Modbus RTU feature
* Data model change
* YML files configurations
* User Guide – Draft version v1.1

**Sprint 5**

* Byte swap and word swap for on-demand read and write for Modbus RTU
* Byte swap and word swap for on-demand read and write for Modbus TCP
* MQTT Export mapping to JSON payload
* Write response mapping in MQTT Export.
  + At present, a common topic is used for both write and read response.
* Configurable response time or both RTU and TCP - stack level configuration
* Remove dependency on boost library
* Added support for a Logger with 3mb log limit: At present, 3 log files are supported on a rotation basis
* An option is added in installation script to make Jenkins job without interaction
* Support is added to avoid deletion of files when uninstall\_EIS script is executed inside a directory other than EIS directory
* Document to apply RT to Ubuntu server

**Sprint 6**

* Write on demand now supports multiple register write request from MQTT
* Correct topic and message format for on-demand read and write request/response for MQTT used
* Multiple RTU slave issue resolved
* KW issues
* Ubuntu server RT patch for testing
* QoS parameter used in data points yml configuration
* Generalized topic name support instead of device-point name and specific topic name
* ETCD removed for yml configuration. docker volume is used to store configurations
* User Guide 1.4 version

**Sprint 7**

* Integration of UWC with EISv2.1 PV release
* Timestamp added in response body to calculate latency in all layer

**Sprint 8**

* Prod mode support for Modbus TCP and Modbus RTU containers
* Prod mode support for MQTT-Export container
* IPC mode for UWC containers
* Adding docker logs size configuration

**Sprint 9**

* Alternative design approach implemented in timer for segmentation fault issues
* Script added to run unit test cases coverage and reports

**Sprint 10**

* Path for realtime and non-realtime operations are separated in MQTT-Export and Modbus containers.
* Following changes are done:
  + Topics on ZMQ are separated. E.g. "TCP1\_PolledData" and "TCP1\_PolledData\_RT".

Please note these are generic topics, mentioned in docker-compose.yml file.

* + For on-demand read/write operations, JSON payload coming from MQTT can have a field as "realtime":

"realtime":"0" or "realtime":"1"  
If this field is not specified, then default behavior is non-realtime.

* Design is updated such that data for all polling points is sent within polling interval.
  + If data for a point is not received within certain time period called as cutoff time (say 90% of polling interval), then a "Bad" response is sent.
  + If response ("Good" or "Bad") is received after cutoff time, that response is sent immediately.
* Value and timestamps of "Good" response are stored from polling operation. This value is used when publishing an error (i.e. "Bad") response for polling operation.
  + If "Bad" scenarios occur without any "Good" result in polling, then "value" field in JSON will be blank. Otherwise, "value" field will have data from last known "Good" scenario.
* Default value of response timeout is 80 msec.

**Sprint 11**

* Thread priority and scheduler are configured using RT patch
* A global configuration setting is defined to be used across containers
* Added "retries" mechanism for timed-out requests
* Added support for prioritizing operations like Polling RT, Polling Non-RT, etc.
* Added support for defining default real time behavior
* QOS parameter to be used by MQTT-Export is now defined in global configuration setting
* A field “lastGoodUsec” is added for responses having "Bad" status
* For on-demand read/write requests, timestamps for request processing within MQTT-Export are also captured
* Parity for RTU communication is indicated by 'N', 'O', 'E' instead of 0, 1, 2 to avoid confusion

**Sprint 12**

* Dockerfile optimization for deployment (UWC-230)
* Changed error code format to uniform representation in JSON model

**Sprint 12.1**

* Change the Time scaling and Time basis in reported data for performance analyses

**Sprint 13**

* UWC Integration with EIS 2.2 PV release
* UWC Installation using EIS Installer
* Integrated Device Manageability framework and ThingsBoard interface using EISv2.2
* User Guide update

**Sprint 13.1**

* Installing UWC using existing scripts instead of EIS Installer.
* User Guide update

**Sprint 14**

* Add “datatype” field in datapoints-YML. Make it available in JSON payload for polling.
* Add “Persistence” field in datapoints-YML. Make it available in JSON payload for polling.

**Sprint 15**

* Clubbing of multiple RTU containers as one
* SCADA RTU Implementation (NDEATH and NBIRTH)
* Latest Paho Library used
* Renaming the field in JSON payload for e.g. (Topic name changed from “topic” to “data\_topic”)
* Simplified enabling ZMQ modes (IPC Prod, IPC Dev, TCP Prod and TCP Dev) for UWC

**Sprint 15.1**

* KW issues resolved

**Sprint 16**

* SCADA RTU Implementation (DBIRTH)
* Secure MQTT connection by enabling TLS mode of communication
* Logs Persistence after restart/reboot of Gateway

**Sprint 17**

* SCADA-RTU-Config.yml for storing configuration for SCADA-RTU
* Non secured communication between SCADA Master (e.g. Ignition) and UWC
* DBIRTH message for vendor app with supported datatype
* DDATA message for vendor app with all supported data types
* Using Ignition as SCADA Master for testing

**Sprint 18**

* SCADA-RTU shall use configured QoS parameter defined for internal and external MQTT communication
* DDEATH message for vendor app
* DBIRTH message for Modbus device with all properties from YML file for all metrics
* Timestamp in DDATA, DBIRTH messages for vendor app
* Explore options for establishing TLS communication between SCADA-RTU and Ignition system
* DCMD message for vendor app with all supported datatypes
* DCMD message for Modbus devices
* START\_BIRTH\_PROCESS from SCADA-RTU

**Sprint 19**

* Secured-TLS/SSL communication between Ignition and UWCP
* Integrate EIS-DBS patch with UWC
* Common Custom network (network name - uwc\_nw) is created for all UWC containers
* Poll update and DDATA message for Modbus devices
* Poll update and DDEATH/DBIRTH message for Modbus devices
* Installation of SCADA-RTU module shall be optional
* scada\_config.yml file shall be generated using user inputs
* MQTT connection break scenarios

**Sprint 20**

* Provide HEALTHCHECK mechanism to address DBS warning
* Explore and create a simple patch mechanism for EIS DBS fix patch
* Analyze DBS reported issues
* KPI Tactic 4 – Application development to identify number of control loops
* KPI Tactic 5 – Application development to identify number of stress points for a control loop
* Review comment: Remove all redundancy in UWC code base (sprint 16 codebase)

**Sprint 21**

* KPI Tactic - Data collection for analysis
* Set default log size for KPI App to 100 MB
* MQTT connection break scenarios
* Use uniform MQTT handling mechanism in SCADA-RTU and MQTT-Export (code base sprint 20)
* Fix DBS report of sprint 20
* Testing of KPI application

**Sprint 22**

* KPI Tactic - Data collection for analysis
* Insert a variable for build number reference in docker-compose

**SupportRelease1**

* UWC License.txt file content for Releases
* Code review comments incorporated in UWC codebase

**SupportRelease2**

* Code review comments to be incorporated in UWC codebase
* Test KPI application

**SupportRelease3**

* Sparkplug template feature support is required for compliance with Sparkplug standard

**Sprint 23**

* This is a build for fixing reported issues.

**Sprint 24**

* Sparkplug template feature is supported
* Design change for Template Feature
* Adding support for a new message of template definition – TemplateDef
* Adding support for changed vendor app message for DBIRTH/DDATA/DCMD

**Sprint 25**

* Test Scada-RTU on the EII2.4 ported UWC code base
* Configuring UWCP and AWS
* Supporting NCMD for Rebirth control

**Sprint 26**

* Non-interactive mode is added for the installation
* Build scripts are organized into a separate folder
* Renamed SCADA-RTU to Sparkplug-Bridge and MQTT-Export to MQTT-Bridge

**Sprint 27**

* Issue fixes

**Sprint 28 and 29**

* Hex data conversion for Polling, On Demand Read and On Demand Write
* Hex data conversion for Sparkplug-bridge

**Sprint 30 and 31**

* Add date and removed EPOCH time from logger for all uwc containers
* modconn stack integrated with uwc
* Sphinx for user guide v1.39
* Rounding off decimal values for Hex Data Conversion
* Updated all the source files (\*.hpp, cpp), shell scripts, dockerfiles, docker-compos files with new MIT license

## Changes to Existing Features

* None

## Unsupported or Discontinued Features

* Removed “persistence” field from datapoints YML file.
* SCADA-RTU no longer publishes msg on existing MQTT.
* Removed support for "isinput" field from datapoints.yml file
* Configuration files to test TCP PROD and TCP DEV mode for ZMQ communication in UWC are removed.
* ETCD UI is no longer installed under UWC.
* Older control loop application and associated references are removed. KPI App should be used going forward.
* Generation of unique name for edge node for SCADA-RTU is removed. User should ensure unique name generation using “group-id” and “edge-node-id” fields.

* Instrumentation logs are disabled in release mode for Modbus and MQTT-Export modules (Support Release\_3).
* Use of ETCD for changing YML files (Release Notes\_R1).
* Use of device-point specific topic names for EIS (Release Notes\_R1).
* JSON mapping file (Release Notes\_R1).
* No Width and Address Parameter support in DBIRTH message as per Template feature

# Fixed Issues

* UWC - update installation instructions to ensure installer is aware of the auto-update and cloud connection options by OS supplier
* Change Scada\_config.yml name to sparkplug-bridge\_config.yml
* Unit Test case failing for modbus container
* Memory leak & Dead code in UWC\_Common library ZmqHandler.cpp
* Not getting CMD message for template features

# Known Issues / Limitations

1. This is a development build and testing is in progress.

# Related Documentation

1. UWC User Guide v1.40

# Where to Find the Release

1. Sources and Documents uploaded on Exxon’s share point

## How to Install this Release (Optional)

Refer Installation section of the UWC User Guide

# Best Known Configuration (optional)

For the latest Client Based Broadwell U Mobile Platforms Best Known Configuration (BKC), go to the Best Known Configuration page on Intel® PC Design Center or on CDI

<https://sharedspaces.intel.com/sites/PCDC/SitePages/Ingredients/ingredient.aspx?ing=BKC>

<http://www.intel.com/cd/edesign/library/asmo-na/eng/537047.htm>

## Tested Development Kits and Configurations (optional)

Intel® Xeon® Processor E3-1125C with Intel® Communications Chipset 8910 Development Kit (Stargo)

# Hardware and Software Compatibility

* 4th Generation Intel® Core™ Processor Platform
* 4th Generation Intel® Core™ Processor U Series-based Platform
* 4th Generation Intel® Core™ Processor Y Series-based Platform
* 3rd Generation Intel® Core™ Processor Platform
* 3rd Generation Intel® Core™ Processor U Series-based Platform
* 3rd Generation Intel® Core™ Processor Y Series-based Platform
* Intel® Pentium® Processor 2000/A1000/G2000/G2100 Series
* Intel® Celeron® Processor 900/1000/G1000 Series
* Intel(R) Pentium(R) N3510/ 3520 Processor
* Intel(R) Celeron(R) N2810/ 2820 Processor
* Intel(R) Celeron(R) N2910/ 2920 Processor
* Intel(R) Celeron(R) N2805/ N2806 Processor
* Intel(R) Celeron(R) N2815 Processor
* Intel(R) Pentium(R) J2850/ J2900 Processor
* Intel(R) Celeron(R) J1850/ J1900 Processor
* Intel(R) Celeron(R) J1750/ J1800 Processor
* Intel(R) Celeron(R) Pentium QC (N3520)1
* Intel(R) Celeron(R) QC (N2920)1
* Intel(R) Celeron(R) DC (N2820)1
* Intel(R) Celeron(R) DC (N2806)1
* Intel(R) Celeron(R) DC (N2815)1
* Intel(R) Celeron(R) QC (J2900)1
* Intel(R) Celeron(R) QC (J1900)1
* Intel(R) Celeron(R) DC (J1800)1

## BIOS/Firmware Version (optional)

Development platform “Stargo” maps to BIOS/Firmware version available in CDI “[Cave Creek] Chipset (embedded) - Stargo CRB BIOS - Rev. 46 (CDI# 494555)”