

EPICS Support for OPC UA – SRS

Software Requirements Specification

Author: Ralph Lange

Status: 1.1 release

Table of Contents

Table of Contents	1
Introduction	3
Scope and Purpose	3
Acronyms and Definitions	3
References	4
Background	4
OPC	4
OPC UA	4
OPC Unified Architecture Overview	4
Sessions	5
Nodes and NodeIds	5
Browse Paths	5
Data Types	5
Basic Operations	5
Subscription Model	6
Parameters	6
Software Requirements Identification and Classification	6
Identifier	6
Category	6
Priority	7
General Description	7
OPC UA Client	7
EPICS Device Support	7
Use Cases	8
EPICS Community Context	8
CODAC Core System Context	8
WinCC OA Server	8
Siemens S7-1500 Series Server	8
Interfaces	8
IOC Interface	8

Device Support	8
Record Support	9
Startup Script	9
Requirements	9
Functional Requirements	9
Sessions and Subscriptions	9
Data Types	10
Operations	12
Health and Status	14
Interface Requirements	14
Configuration	16
Performance Requirements	17
Other Requirements	17
Context	17
Debugging	18
Quality	18
Documentation	19

Introduction

Scope and Purpose

This document is gathering requirements for the development of EPICS Device Support for OPC UA, a component that establishes communication between an EPICS IOC Database and an OPC UA server.

It tries to follow IEEE 830 and best practices in software engineering.

The intended audience of this document includes stakeholders on one side and software architects, developers and testers on the other.

Acronyms and Definitions

AC	Alarm and Conditions (part 9 of OPC UA spec)
CA	Channel Access
CCS	CODAC Core System
CIS	Central Interlock System
CLI	Command line interface
DA	Data Access (part 8 of OPC UA spec)
EPICS	Experimental Physics and Industrial Control System
HA	Historical Access (part 11 of OPC UA spec)
IOC	Input output controller
OLE	Object Linking & Embedding
OPC	Open Platform Communications (OLE for Process Control)
OPC UA	OPC Unified Architecture
PLC	Programmable logic controller
PV	Process variable
SRS	Software requirements specification
XML	Extensible Markup Language

References

- [1] "OPC Foundation." <https://opcfoundation.org/>. Accessed 9 Feb. 2018.
- [2] "C++ Based OPC UA Client/Server SDK: Main Page - Documentation" <http://documentation.unified-automation.com/uasdkcpp/1.5.5/html/index.html>. Accessed 9 Feb. 2018.
- [3] "EPICS Application Developer's Guide." 26 Jan. 2001, <https://epics.anl.gov/base/R3-16/1-docs/AppDevGuide.pdf>. Accessed 23 Feb. 2018.
- [4] "CODAC Core System - ITER." <https://www.iter.org/mach/codac/CoreSystem>. Accessed 27 Feb. 2018.

Background

OPC

The OPC standard is a series of specifications developed by industry vendors, end-users and software developers. These specifications define the interface between Clients and Servers, as well as Servers and Servers, including access to real-time data, monitoring of alarms and events, access to historical data and other applications.

When the standard was first released in 1996, its purpose was to abstract PLC specific protocols (such as Modbus, Profibus, etc.) into a standardized interface allowing HMI/SCADA systems to interface with a "middle-man" who would convert generic-OPC read/write requests into device-specific requests and vice-versa.

Initially, the OPC standard was restricted to the Windows operating system. As such, the acronym OPC was borne from OLE (object linking and embedding) for Process Control. These specifications are now known as OPC Classic. [1]

OPC UA

The OPC Unified Architecture (UA), released in 2008 by the OPC Foundation, is a platform independent service-oriented architecture that integrates all functionality of the individual OPC Classic specifications into one extensible framework, providing

- *Functional equivalence*: all COM OPC Classic specifications are mapped to UA
- *Platform independence*: from embedded to cloud-based infrastructure
- *Security*: encryption, authentication, and auditing
- *Extensibility*: ability to add new features without affecting existing applications
- *Information modeling*: for defining complex information. [1]

OPC Unified Architecture Overview

A more detailed overview of OPC UA concepts can be found in [2].

Sessions

A connection between an OPC UA client and a server is called *Session*. OPC UA offers certificate based authentication of client and server, and transport security by enforcing encryption and signing of network messages.

Nodes and NodeIds

In the OPC UA communication model, every entity in a server's address space is a *Node*. Each node has a unique *NodeId* to identify it, which always consists of three elements:

NamespaceIndex

The server-side index into a table of namespaces (URIs), where each namespace URI identifies the namespace naming authority. This allows multiple namespaces to coexist without interfering on a single server. Namespace indexes are numeric for performance reasons.

IdentifierType

The format and data type of the identifier. Can be string, number, globally unique identifier (GUID), or a namespace specific opaque value.

Identifier

The node's identifier.

Browse Paths

Another way of specifying a node in OPC UA address space is using a *Browse Path*. The browse path is a sequence of node names, starting from a known location (e.g. the root of the OPC address space). The browse path can also be expressed as a string, where the individual node names are separated by delimiters (e.g. "/" to express hierarchy).

Browse paths add a layer of abstraction that can be used to provide different views of the address space or to support more than one naming convention.

In the end, each item must be addressed by its *NodeId*. The OPC UA server offers services to resolve a browse path to a *NodeId* as necessary.

Data Types

OPC UA uses a set of 24 built-in types, in both scalar and array form. This includes a Null type, Boolean, signed and unsigned integers (8, 16, 32 and 64 bit wide), single and double precision floats, strings, a time stamp and several types for OPC UA specifics. Data can also be of the type *ExtensionObject*, which translates either to a variant (union) of the built-in types, or to a structure, where each element can again be either a built-in type or an *ExtensionObject*.

Basic Operations

As basic operations for accessing variables, OPC UA supports read and write services, for an arbitrary number of nodes per call. Both blocking and non-blocking interfaces are defined.

Subscription Model

In addition to polling (periodically reading), OPC UA provides a so-called *Subscription*, for which a client subscribes to a selection of nodes and lets the server monitor these items. In case of changes the client is notified. This mechanism greatly reduces the amount of transferred data and is the recommended mechanism to “read” information from a UA Server.

A client may subscribe to different types of information provided by an OPC UA server, e.g. variable values, events and aggregated data. A subscription groups these sources of information, called *Monitored Items*, together. The server sends all updates from a subscription periodically to the client, as a *Notification*.

A subscription has to be created within the context of a *Session*.

Parameters

There are multiple settings available for monitored items and the subscription.

The *sampling interval* defines, individually for each monitored item in the subscription, the rate at which the server checks the data source for changes. It can be set faster than the notification to the client, in which case the server may queue the samples and publish the complete queue (see below).

Filters contain criteria to distinguish which data changes or events should be reported and which should be blocked.

A UA server may support queuing of data samples or events. The *queue size* can be configured for each monitored item. When the data gets delivered to the client, the queue is emptied. Each queue has a *discard policy* in case an overflow occurs.

By setting the *monitoring mode*, data sampling and reporting for a monitored item can be enabled or disabled.

Two settings affect the subscription. After each *Publishing Interval*, notifications collected in the queues are delivered to the client in a *Notification Message*. Sending of notification messages can be enabled or disabled using the *Publish Enabled* setting.

Software Requirements Identification and Classification

Identifier

The Computer Software Configuration Item (CSCI) identifier is OPCUA. The requirements identifiers are formatted as:

OPCUA-SRS-<req-category-id>-<req-number>

Category

The requirement categories used in this document are:

F	Functional
I	External Interface

P	Performance
O	Other

Priority

The *Priority* defines the importance of the requirement, which has implications on the implementation.

- **High** implies that any implementation not addressing this requirement is incomplete.
- **Normal** means that the requirement shall be addressed in a version of the implementation, but a specific software version may not cover it.
- **Low** implies the implementation is optional.

The following transient values may also be used:

- **TBD**: unstable requirement that cannot be qualified.
- **Cancelled**: suppressed requirement. (Only for traceability of rejected requirements.)

General Description

OPC UA is a generic, versatile, flexible protocol for the integration of a wide variety of industrial components. To maximize usability and reusability, the EPICS Device Support should try to be as flexible and generic, supporting all OPC UA data types, offering options to fine-tune function and performance to adapt to all kinds of use cases. The goal is to give the user freedom to choose the right granularity and parameters, and support them in finding the right compromise between performance, simplicity and adaptation.

OPC UA Client

The EPICS Support for OPC UA module is expected to use an available library to implement the OPC UA client functionality.

EPICS Device Support

The Device Support is connecting the EPICS database to the mentioned client library. Its main functions are:

- To allow configuring all necessary attributes and properties to set up the OPC UA connections.
- To appropriately map the EPICS database processing model to the operations provided by the client library API.
- To provide status and health information about the OPC UA connection.

Use Cases

EPICS Community Context

The EPICS Support for OPC UA module is targeted for wider use in the EPICS community, to achieve a broader set of usage that will increase its quality and robustness.

CODAC Core System Context

This module is scheduled to be part of the CODAC Core System 6 release series [4], which is based on RedHat Enterprise Linux 7.4 and EPICS 7.0.

WinCC OA Server

In the context of the ITER Central Interlock System (CIS), a relatively small set of signals is generated within the central WinCC OA application. They will be delivered to the control system as non-critical signals, interfaced using the OPC UA server that is embedded in the WinCC OA application.

- Signals: few. (<50)

- Read-only access.

- Update period: short. (50-100ms)

Siemens S7-1500 Series Server

The Siemens S7-1500 series of PLCs offers an embedded OPC UA server, running on the PLC itself. Several ITER Plant Systems have shown interest in using OPC UA to integrate this type of PLCs.

The first and probably most demanding use case is the integration of the building automation systems. These systems will include user defined structures on the PLC, which in the OPC UA server are mapped to ExtensionObjects of structure type.

- Signals: many. (>50.000)

- Read and write access (including structured data).

- Update period: medium. (200ms)

Interfaces

IOC Interface

Device Support

For a Classic EPICS Device Support, in addition to the implementation, the support module provides a function table of type DSET for each record type it supports. [3]

Configuration is done through the EPICS database input or output links. Additional optional parameters can be supplied through so-called info items.

Device Support modules are declared through a Database Definition (DBD) snippet.

Record Support

If a module implements an EPICS record type, in addition to the code it provides a function table of type RSET, and the complete record declaration (including documentation) as a Database Definition (DBD) snippet. [3]

Startup Script

In most cases, low level setup and control of debugging features of an EPICS Device Support is done through calls from the startup script (st.cmd) that are interpreted by the IOC's command line interpreter *iocShell*. A module providing iocShell commands must declare them properly and add an initialization hook routine in its Database Definition (DBD) snippet. [3]

Requirements

Functional Requirements

Sessions and Subscriptions

OPC-SRS-F-005	Support TCP UA Binary transport
Priority Milestone	High 0.7
Requirement	The module shall be able to connect to OPC UA servers using the TCP UA Binary transport.
Remarks	TCP is the most commonly used transport for OPC UA. The web service transport seems less appropriate for control system use.

OPC-SRS-F-010	Support OPC UA Security
Priority Milestone	Normal
Requirement	The module shall support OPC UA security, i.e. certificate based authentication and transport encryption.

OPC-SRS-F-015	Allow multiple sessions
Priority Milestone	Normal
Requirement	The module shall be able to connect to multiple OPC UA servers by setting up multiple sessions.

Remarks	As Linux IOCs don't use a lot of resources, one connection per IOC is equally appropriate.
---------	--

OPC-SRS-F-020	Deferred connection to server
Priority Milestone	High 0.7
Requirement	The module shall be able to transparently handle the case of the server not being available when the IOC starts.

OPC-SRS-F-025	Automatic reconnection handling
Priority Milestone	High 0.7
Requirement	In case of connection loss, the module shall provide a mode in which it periodically tries to reconnect and recovers completely after a successful reconnection.
Remarks	The mode (automatic/manual reconnection) and the reconnection period shall be configurable for each session. CLI and error logs shall show disconnect and reconnect as events. Unsuccessful reconnection attempts shall no be logged (unless when debugging)

OPC-SRS-F-030	Properly shut down connection on IOC reboot
Priority Milestone	High 0.7
Requirement	On IOC reboot, the module shall properly unregister and shut down any open OPC UA subscriptions and sessions.
Source	UC S7-1500 : improper shutdown has shown to cause performance degradation on the server.

OPC-SRS-F-035	Support multiple subscriptions per session
Priority Milestone	High 1.0
Requirement	The module shall be able to create multiple subscriptions per OPC UA session.
Source	UC S7-1500 : when dealing with many items, multiple subscriptions allow separation of high priority signals (small publishing interval) from slow, low priority signals (larger publishing interval).

Data Types

OPC-SRS-F-105	Support basic OPC UA data types
Priority Milestone	High 0.7

Requirement	The module shall be able to handle OPC UA data of all basic OPC UA data types: Null, Boolean, SByte, Byte, Int16, UInt16, Int32, UInt32, Int64, UInt64, Float, Double, String.
-------------	---

OPC-SRS-F-110	Support arrays of basic OPC UA data types
Priority Milestone	High 0.7
Requirement	The module shall be able to handle OPC UA array type data of all basic OPC UA data types: Null, Boolean, SByte, Byte, Int16, UInt16, Int32, UInt32, Int64, UInt64, Float, Double, String.

OPC-SRS-F-115	Support OPC UA ExtensionObject structures
Priority Milestone	High 0.7
Requirement	The module shall be able to handle OPC UA data structures, i.e. ExtensionObject type data that encodes a structure. Nested structures shall be supported for a depth of at least 10.
Source	UC S7-1500 : when dealing with many items, user defined structures are much more performant (100x) than single subscriptions.

OPC-SRS-F-120	Support OPC UA ExtensionObject variants
Priority Milestone	Low
Requirement	The module shall be able to handle OPC UA variants, i.e. ExtensionObject type data that encodes a variant.

OPC-SRS-F-125	Range checks when converting data
Priority Milestone	High 1.0
Requirement	The module shall apply range checks when converting data between EPICS records and set record status accordingly.

OPC-SRS-F-130	Handle server side type changes on reconnect
Priority Milestone	High 1.0
Requirement	The module shall transparently handle cases where the OPC UA data type of an item changes across a server reboot.
Remarks	Intended to make development easier and more robust.

Operations

OPC-SRS-F-205	Limit number of items per OPC UA operation
Priority Milestone	High 0.7
Requirement	The module shall allow to set a limit on the number of items used per OPC UA service call.
Source	UC S7-1500 : The server has such limitations.

OPC-SRS-F-210	Support OPC UA write operation for output records
Priority Milestone	High 0.7
Requirement	The module shall support issuing an OPC UA write operation on a Node when an output record is being processed.
Remarks	Since the OPC UA operation may take time, record processing must be asynchronous.

OPC-SRS-F-215	Support OPC UA read operation for input records
Priority Milestone	High 1.0
Requirement	The module shall support issuing an OPC UA read operation on a Node when an input record is being processed.
Remarks	Since the OPC UA operation may take time, record processing must be asynchronous.

OPC-SRS-F-220	Support OPC UA monitor for input records
Priority Milestone	High 0.7
Requirement	The module shall support setting up a Monitored Item on an OPC UA subscription for an input record and processing the record whenever a data update is received.

OPC-SRS-F-225	Support OPC UA monitor for output records
Priority Milestone	High 0.7
Requirement	The module shall support setting up a Monitored Item on an OPC UA subscription for an output record and updating the record whenever a data update is received.

OPC-SRS-F-230	Multi-record monitors for OPC UA data structures
Priority Milestone	High 0.7

Requirement	The module shall support setting up a single Monitored Item on an OPC UA subscription for an OPC UA data structure, and allow multiple input and output records to bind to leaf elements of the structure. Whenever a data update for the structure is received, all record data shall be updated and the records processed accordingly.
Source	UC S7-1500 : see OPC-SRS-F-115 .

OPC-SRS-F-235	Writing of OPC UA data structures
Priority Milestone	High 0.7
Requirement	The module shall support multiple output records to bind to leaf elements of a single OPC UA structure. When a specific record is being processed, current data from all records connected to the structure shall be collected and written using a single OPC UA write operation.
Source	UC S7-1500 : writing a complete set of configuration parameters.
Remarks	Since the OPC UA operation may take time, record processing must be asynchronous.

OPC-SRS-F-240	Support queuing of data samples for Monitored Items
Priority Milestone	Normal
Requirement	The module shall support handling of server side queueing of data samples for Monitored Items, with configurable queue size.
Source	UC S7-1500 : efficiency may require server side queueing on subscriptions with a long publish interval.
Remarks	Server side queues will require matching client side queues to avoid losing updates.

OPC-SRS-F-245	Support filters for Monitored Items
Priority Milestone	Low
Requirement	The module shall support handling of server side filters for Monitored Items.
Source	UC S7-1500 : efficiency may require server side filters.

OPC-SRS-F-250	Support OPC UA server and source timestamps
Priority Milestone	High 1.0
Requirement	The module shall support using the OPC UA supplied server timestamp or source timestamp for the EPICS record.

OPC-SRS-F-252	Support reading timestamp from structured data
Priority Milestone	High 1.0
Requirement	The module shall support using a configurable top level element of a data structure as timestamp for EPICS records connected to the structure.

OPC-SRS-F-255	Support reading, writing and monitoring array subranges
Priority Milestone	Normal
Requirement	The module shall support OPC UA read, write and monitor operations on subranges of arrays.

OPC-SRS-F-260	Support calling OPC UA Methods
Priority Milestone	High 1.0
Requirement	The module shall support calling OPC UA Methods, supplying a list of arguments and receiving lists of argument error codes and return values.

Health and Status

OPC-SRS-F-305	Access to OPC UA Status Codes
Priority Milestone	Low
Requirement	The module shall allow access to the OPC UA Status Code related to an item.
Remarks	OPC UA defines hundreds of status codes. The standard EPICS record capabilities are not allowing a sufficient representation. Both the numerical and text representation should be available.

OPC-SRS-F-310	Provide health and statistics data
Priority Milestone	High 1.0
Requirement	The module shall provide health and statistics data on session, subscription and item level, including but not limited to: Latest status code (item), connection status (session), number of items (session, subscription), number of reads/writes (session, item), number of updates (subscription, monitored item), readback of configuration parameters (session, subscription, monitored item).
Source	ITER Health Monitoring Requirements.

OPC-SRS-F-315	Support Subscriptions to OPC UA Events
Priority Milestone	Normal

Requirement	The module shall provide support for subscribing to OPC UA Events.
Remarks	OPC UA Events provide out-of-band information, which is especially useful in set-ups with OPC UA server and device being separate.

Interface Requirements

OPC-SRS-I-005	Support all applicable record types from EPICS Base
Priority Milestone	Normal 1.0
Requirement	The module shall support all applicable record types from EPICS Base.

OPC-SRS-I-010	Configurable reconnection attempt period
Priority Milestone	Normal 1.0
Requirement	In automatic reconnection handling mode (OPC-SRS-F-025), the period between attempts to reconnect shall be configurable for each session.

OPC-SRS-I-015	Configurable item batch size for OPC UA services
Priority Milestone	High 0.7
Requirement	The module shall use a configurable maximum number of items per call to OPC UA services.
Source	UC S7-1500 : The server has limitations on the maximum number of items per operation.

OPC-SRS-I-020	Simple single record configuration
Priority Milestone	High 1.0
Requirement	For a “simple” mapping of a standard record to an OPC UA item of basic type, the configuration shall be possible inside that single record.

OPC-SRS-I-025	Use of addressing standards
Priority Milestone	High 0.7
Requirement	For all configuration, the module shall follow existing standards and conventions, if applicable.

OPC-SRS-I-030	Configurable session connect info
Priority Milestone	Normal
Requirement	The module shall support configuration of the session info that is used when

	connecting to the server.
Remarks	The session connect info allows to identify the client of a session on the server side, which is convenient during debugging.

OPC-SRS-I-035	Support Namespace definition through URI
Priority Milestone	Normal
Requirement	The module shall support configuration of namespaces through their URIs.
Remarks	Following the OPC UA specification, the numerical namespace index may change between server reboots. Configuring the URI and getting the namespace index from the server is the fully compliant behavior.

OPC-SRS-I-040	Support Node definition through a Browse Name
Priority Milestone	Normal
Requirement	The module shall support configuration of items through Browse Names.

OPC-SRS-I-045	Subscription runtime control
Priority Milestone	Normal
Requirement	For a subscription, publish interval and publish enable switch shall be changeable at run time.

OPC-SRS-I-050	Monitored Item runtime control
Priority Milestone	Normal
Requirement	For a Monitored Item, the sampling interval and the Monitoring Mode shall be changeable at run time.

Configuration

OPC-SRS-I-110	Configurable readback update for output records
Priority Milestone	High 1.0
Requirement	The mechanism to set up a monitor and update an output record when the value changes on the server (OPC-SRS-F-225) shall be configurable for each record, with a selectable default choice.

OPC-SRS-I-115	Configurable Monitored Item properties
Priority Milestone	High 0.7
Requirement	The item specific properties for a Monitored Item (see OPC-SRS-F-220)

	shall be configurable, with a configurable default for each.
--	--

OPC-SRS-I-120	Configurable timestamp handling
Priority Milestone	High 1.0
Requirement	For each record, the timestamp used for EPICS record processing (see OPC-SRS-F-250 , OPC-SRS-F-252) shall be selectable between IOC timestamp, OPC UA server timestamp, OPC UA source timestamp and (for structured data) the data timestamp (element configurable per item).

OPC-SRS-I-125	Configurable OPC UA read mechanism for input records
Priority Milestone	Normal
Requirement	The read mechanism (single read OPC-SRS-F-215 or monitor OPC-SRS-F-220) shall be configurable for each record, with “monitor” being the configurable default choice.

OPC-SRS-I-130	Configurable initial read after reconnection behavior
Priority Milestone	Normal
Requirement	The behavior of the initial read operation after the reconnection of a session shall be configurable for each record, with “read” (read value into the record) being the default choice. Other choices shall include “ignore” (do not change record) and, in the case of output records, “write” (overwrite with record values).

Performance Requirements

OPC-SRS-P-005	Large number of variables in a single structure
Priority Milestone	High 1.0
Requirement	The module shall be able to handle a sustained load of 200ms notifications for a single 32k Byte user defined structure linked to 32k long input records.
Source	UC S7-1500
Remarks	The IOC should run on an appropriate host, i.e. 4 core VM.

OPC-SRS-P-010	Large number of Monitored Items
Priority Milestone	High 1.0
Requirement	The module shall be able to handle a sustained load of 200ms notifications for 2k individual float data items linked to 2k analog input records.
Source	UC S7-1500

Remarks	The IOC should run on an appropriate host, i.e. 4 core VM.
---------	--

Other Requirements

Context

OPC-SRS-O-005	Support EPICS Base 3.15.5 (and up)
Priority Milestone	High 0.7
Requirement	The module shall compile and work with EPICS Base 3.15.5 and later releases.
Source	UC Community Context.
Remarks	EPICS Base 3.15 is the <i>stable</i> release series.

OPC-SRS-O-010	Support operating system RHEL 7 (and up)
Priority Milestone	High 0.7
Requirement	The module shall compile and work on RHEL 7.4 systems and later RHEL releases.
Source	UC CCS Context.
Remarks	RHEL is a very conservative distribution – support on this platform should cover CentOS/SL/Fedora as well as Debian/Ubuntu.

OPC-SRS-O-015	Support x86 (64bit) processor architecture
Priority Milestone	High 0.7
Requirement	The module shall compile and work on an x86 (64 bit) architecture.
Source	UC CCS Context.

OPC-SRS-O-020	Extensibility for adding more or other client library options
Priority Milestone	Normal
Requirement	The module shall be designed in a way that allows integration of other or additional OPC UA client libraries without changing the EPICS database.
Remarks	Requiring changes in the startup script should be avoided if possible.

Debugging

OPC-SRS-O-105	Support CLI Debug Tracing
----------------------	----------------------------------

Priority Milestone	High 1.0
Requirement	The module shall provide means to enable CLI tracing for specific sessions, subscriptions and items. The verbosity shall be configurable.

OPC-SRS-O-110	Support CLI Debug Inspection
Priority Milestone	High 1.0
Requirement	The module shall provide CLI functions to dump information about specific sessions, subscriptions and items. The verbosity shall be configurable.

Quality

OPC-SRS-O-205	Design and coding guidelines
Priority Milestone	High 1.0
Requirement	The module shall follow best practices for modularity and design simplicity. It shall undergo design and code review.

OPC-SRS-O-210	EPICS Device Support guidelines
Priority Milestone	High 0.7
Requirement	The module shall follow the guidelines provided in [3].

OPC-SRS-O-215	Unit tests
Priority Milestone	High 1.0
Requirement	The module shall provide unit tests.

OPC-SRS-O-220	Software quality
Priority Milestone	High 1.0
Requirement	The module development shall follow the procedures of the ISO/IEC 12207 Development phase.

Documentation

OPC-SRS-O-305	Design documentation
Priority Milestone	High 1.0
Requirement	The module shall provide requirements and design documentation.

OPC-SRS-O-310	User manual
Priority Milestone	High 1.0
Requirement	The module shall provide a user manual that explains how to use the OPC UA Device Support in a standard EPICS context.