ROBOTICS

Application manual

IRC5 OPC UA Server

Trace back information:
Workspace Main version a390
Checked in 2021-03-01
Skribenta version 5.4.005

Application manual
IRC5 OPC UA Server

Document ID: 3HAC074394-001
Revision: B

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damage to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

Table of contents

This page is intentionally left blank

Overview of this manual

About this manual

This manual contains instructions for daily operation of ABB IRC5 OPC UA Server.

Usage

This manual should be used during operation, installation and configuration of ABB IRC5 OPC UA Server.

Who should read this manual?

This manual is intended for:

- Users of the product ABB IRC5 OPC UA Server.

Prerequisites

The reader should.

- use the manual as an online help and
- have OPC UA Server installed.

References

Reference	Document ID
Technical reference manual - System parameters	*3HAC050948-001*
Technical reference manual - RAPID kernel	*3HAC050946-001*
Operating manual - IRC5 Integrator's guide	*3HAC050940-001*

Revisions

Version	Description
A	First edition.
B	Released with the version 1.0.1. Following updates are made in this revision: • Added the Appendix, *Appendix C - Robotics companion specification on page 73*. • Updated the section, *IRC5 Information Model on page 47*.

Product documentation

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.

Tip

All documents can be found via myABB Business Portal, *www.abb.com/myABB*.

Product manuals

Manipulators, controllers, DressPack/SpotPack, and most other hardware is delivered with a **Product manual** that generally contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with corresponding figures (or references to separate spare parts lists).
- References to circuit diagrams.

Technical reference manuals

The technical reference manuals describe reference information for robotics products, for example lubrication, the RAPID language, and system parameters.

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, software).
- How to install included or required hardware.
- How to use the application.
- Examples of how to use the application.

Continues on next page

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and troubleshooters.

This page is intentionally left blank

1 Getting started

Overview

The ABB IRC5 OPC UA Server displays OPC UA items for robots with configured aliases. Use ABB IRC5 OPC UA Server Configuration tool to create aliases for the robots you wish to monitor. If aliases are not created before the launch of browser, then no items are displayed in OPC UA address space.

Following are the prerequisite for creating aliases:

- First, the robot must be connected to the same computer network in which OPC UA server is running.
- Second, the ABB IRC5 OPC UA Server displays data only for ABB IRC5 robots with configured aliases (a descriptor that identifies a particular robot; see *Aliases on page 21*). Configuration entries are stored in the *The OPC Alias Configuration file*.
- Third, the system running on the controller that the alias refers to must have the PC Interface RobotWare option installed. Starting from RobotWare 6.11, the OPC UA Server RobotWare option is also required. Otherwise, the alias will still be created, but no communication with the controller will be possible.

Note

For details about accessing the IRC5 OPC UA Server from a client over the network, see *Security on page 52*.

How to setup alias

Use the following procedure to setup and view data from an ABB IRC5 robot controller.

Step	Action
1	Create aliases for those robots you want to connect to the IRC 5 OPC UA Server. For more details, see *How to add new robot aliases on page 35*.
2	OPC UA clients can use the discovery process to find ABB IRC5 OPC UA Server over the network for the session connection. Or Can use direct Endpoint URL to establish a connection.
3	Install any OPC UA client to browse the address space, create subscriptions and monitored items, to read and write data. **Note** There are number of free OPC UA Client test tools available for download on Internet, e.g. Softing's dataFEED OPC UA Client.

Product requirements

Overview

Before installing the IRC5 OPC UA Server, ensure that the computer meets the following hardware and software requirements.

Continues on next page

Continued

Hardware

Medium to high-performance industrial or desktop PC, with the following requirements:

Part	Requirement
CPU	2.0GHz or faster processor, multiple cores recommended
Memory	2 GB minimum. 4GB or more recommended
Disk	2+ GB free space, solid state drive (SSD) recommended.Disk

Software

Operating system	Description
Windows 10 Anniversary Update or later	64-bit edition

It is recommended to run Windows updates to get the latest updates to Windows before installing and running IRC5 OPC UA Server. It is also strongly recommended to keep the Windows operating system updated with the latest security updates according to Microsoft recommendation or company policy.

Windows Firewall can block certain features that are necessary to run IRC5 OPC UA Server, which must be unblocked as required. For more information on Windows Firewall, visit www.microsoft.com

RobotWare software requirements

For all RobotWare 6 versions starting from version 6.11, both the “616-1 PC Interface” and the "1582-1 OPC UA Server” RoboWare options are required.

Earlier versions of RobotWare 6 only require the “PC Interface” RobotWare option.

Firewall settings

The firewall settings are applicable to real and virtual controllers. The following table describes the necessary firewall configurations:

Name	Action	Direction	Protocol	Remote Address	Local Service	Remote Service	Application
RobNetscanHost	Allow	Out	UDP/IP	Any	Any	5512,5514	robnetscanhost.exe
IRC5Controller	Allow	IN	UDP/IP	Any	5513	Any	robnetscanhost.exe
RobComCtrlServer	Allow	Out	TCP/IP	Any	Any	5515	robcomctrlserver.exe
IRC5 OPC UA Server	Allow	IN	OPC UA	Any	61510 (configurable)	Any	ABB.Robotics.IRC5.OPCUA.Server

Product features

ABB IRC5 OPC UA Server implements the functionality of the UA Address Space Model 1.04 Specification. It is a UA server which enables UA clients to browse the address space, create subscriptions and monitor items, and read and write data.

Certificates

Certificates are used to establish secure communication between the OPC UA Client and OPC UA Server.

Continues on next page

Endpoints

OPC UA Clients can connect to the ABB IRC5 OPC UA Server using the following URL:

opc.tcp://HOSTNAME:PORT NUMBER/ABB.IRC5.OPCUA.Server

- **HOSTNAME**: We get it using command hostname on command prompt.
- **PORT NUMBER**: Port number configured in OPC UA Config tool. Defult value is 61510.

Example: *opc.tcp://XXXXXX:61510/ABB.IRC5.OPCUA.Server*

User authentication

ABB IRC5 OPC UA Server supports the following user authentication modes:

- Anonymous: No user identity is provided.
- UserName: A user identified by user name and password.

Client Authentication

The ABB IRC5 OPC UA Server authenticates (Identifies) OPC UA client using **Client application instance certificate**.

Client certificates will be stored in certificate store: <C:\ProgramData\ABB\IRC5 OPC UA\CertificateStores>.

Security

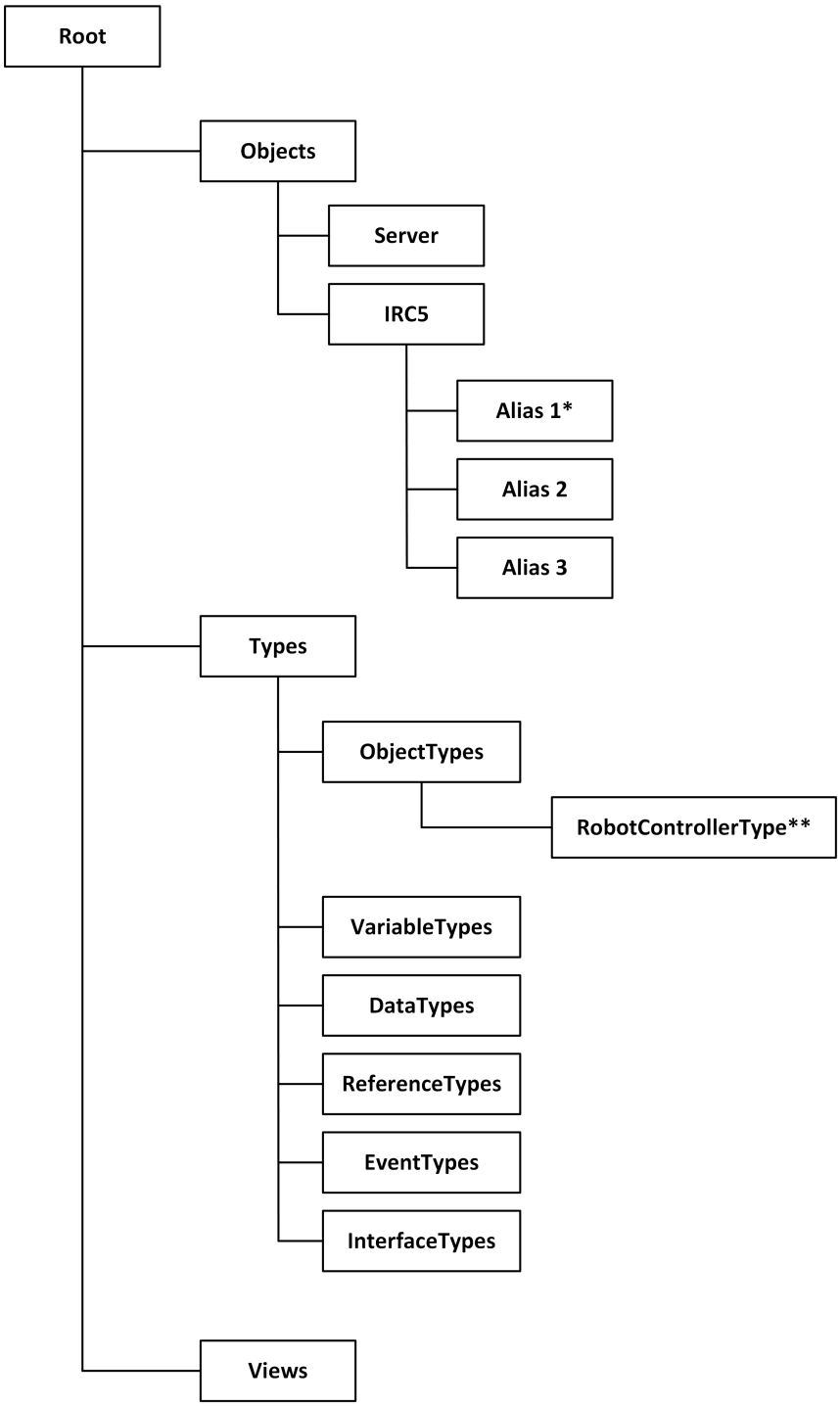
ABB IRC5 OPC UA Server supports OPC UA standard security modes and policies.

Continues on next page

Continued

Address Space

The server's address space represents its contents as a set of Nodes connected by References. The address space begins with the top **node: Root**, which Organizes Objects, Types and Views.



***A robot controller is identified by its alias name that must be unique.**

****Top level object type for an ABB Robotics Controller.**

Continues on next page

Note

For more information on address space, see section *Address space on page 47*.

Cybersecurity

This product is designed to be connected to and to communicate information and data via a network interface. It is your sole responsibility to provide, and continuously ensure, a secure connection between the product and to your network or any other network (as the case may be).

You shall establish and maintain any appropriate measures (such as, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of anti-virus programs, etc) to protect the product, the network, its system and the interface against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Ltd and its entities are not liable for damage and/or loss related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

For more information see section Cyber security for IRC5 robot networks in *Operating manual - IRC5 Integrator's guide*.

Product installation

Prerequistes

The following are the prerequisites to install ABB IRC5 OPC UA Server:

- Administrative rights are required in order to install and configure IRC5 OPC UA Server.
- Install latest ABB Robot Communication Runtime from *http://developercenter.robotstudio.com/landing*

Procedure

Use the following procedure to install IRC5 OPC UA Server:

1. Click Setup.exe from *http://developercenter.robotstudio.com/landing*.

   The IRC5 OPC UA Server Setup window opens.
2. Proceed to install IRC5 OPC UA Server.

   IRC5 OPC UA Server is installed.

This page is intentionally left blank

2 ABB IRC5 OPC UA Server configuration application

2.1 Introduction

Overview

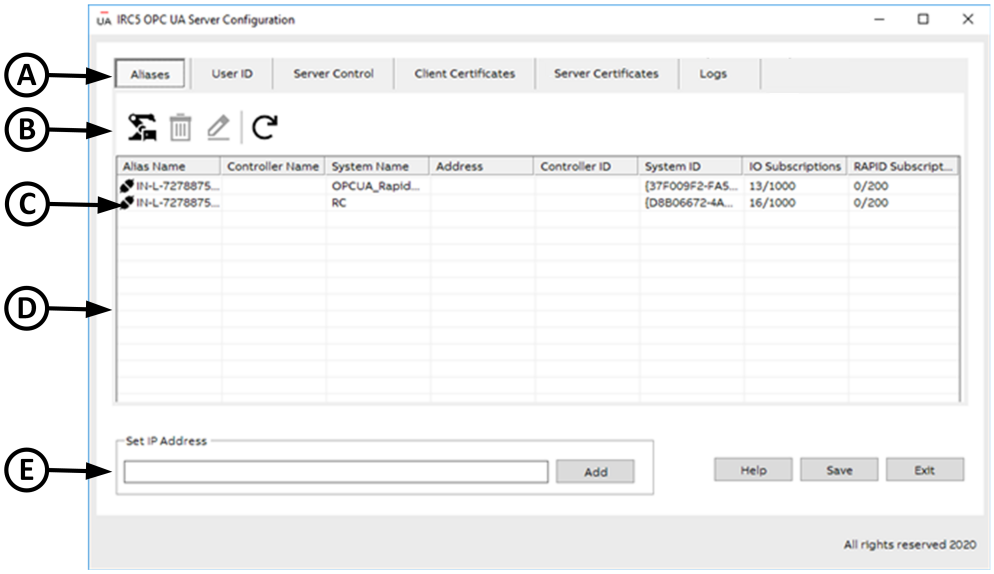
The ABB IRC5 OPC UA Server configuration application is used to create and manage Aliases for ABB IRC5 robot controllers. An Alias is a user-friendly descriptor that represents a communications interface with ABB IRC5 robot controller. You need to create an alias for each robot controller that will be accessed by the ABB IRC5 OPC UA Server.

2.2 About the ABB IRC5 OPC UA Server configuration application

2.2.1 Main screen components

Server configuration

The ABB IRC5 OPC Server Configuration application main screen shows a list of Aliases that you have created. The main screen of ABB IRC5 OPC Server Configuration application displays important information about the created Aliases, such as assigned name, Controller Name, System Name, Address, and so on as shown in the following figure:



xx2000000136

A	Function tab
B	Toolbar buttons
C	Device status icons
D	Device pane
E	Add remote controller

Function tabs

The following table provides information about the function tabs in ABB IRC5 OPC UA Server Configuration application:

Component	Function
Alias	The ABB IRC5 OPC UA Server Configuration application "main screen".
User ID	Enter the user name and password for OPC UA Server access to the robot. For more information, see *OPC UA Server username/password on page 32*.
Server Control	Select the Start or Stop button to start /stop the IRC5 OPC UA Server.
Client Certificates	Provides an interface for managing the configuration of the security certificates for the application.

Continues on next page

Component	Function
Server Certificates	Provides an interface to create a new application certificate, add the application to Local Discovery Server (LDS) trust list, Import application certificate, & Export application certificate.
Logs	Lists all the information/error/warning messages related to OPC UA Server.

Toolbar buttons

The following table provides information about the tool bar buttons in the main screen:

Button	Function
 Add Alias	Opens the Add IRC5 Robot Alias dialog see *Add New Alias dialog screen components on page 24*.
 Delete Alias	Deletes the highlighted alias from the device pane.
 Edit Alias	Opens the Edit Alias dialog see *Edit Alias dialog components on page 27*.
 Refresh Main Screen	Click to refresh Alias connectivity status.

Device pane

The device pane displays a list of robot aliases and their associated attributes.

Component	Function
Alias Name	Displays the name of the alias.
Controller Name	The name of the IRC5 controller.
System Name	The RobotWare system name running on the IRC5 controller.
Address	The IP-address of the IRC5 controller.
Controller ID	The ID of the IRC5 controller.
System ID	The ID of the RobotWare system running on the IRC5 controller.
IO Subscriptions	The number of I/O signal "change-events" subscribed to the IRC5 controller (currently subscribed / maximum number of subscriptions allowed). **Note:** This value is only updated when a Refresh button is pressed.

Continues on next page

2.2.1 Main screen components
Continued

Component	Function
RAPID Subscriptions	The number of RAPID variable "change-events" subscribed to the IRC5 controller (currently subscribed / maximum number of subscriptions allowed). **Note:** This value is only updated when a Refresh button is pressed.

Device status icons

The following table provides information about the device status icons used in the main screen:

Icon	Description
 Connected Alias	Alias is connected and communicating over the network.
 Disconnected Alias	Alias is disconnected from the network.
 Unaccessable Alias	Alias is connected to the network, but the PC Interface and/or the OPC UA Server RobotWare option is missing.

2.2.2 Aliases

Overview

To define an Alias, you need to associate it with one or more of the parameters. Alias and parameters and how to use them are described in this section.

Aliases

An Alias is a user-friendly descriptor that represents a communications interface to a single device such as an ABB robot controller. You need to create Aliases for each robot controller that will be accessed by an application such as the ABB IRC5 OPC UA Server.

Applications such as the IRC5 OPC UA Server use Aliases to address a particular robot controller on a network, so that the application can write data to the controllers and receive data from the controllers.

Parameters

When an IRC5 robot controller is connected to a network, it identifies itself by broadcasting certain information onto the network. This information includes the following parameters:

- Controller Name
- Controller ID
- System Name
- System ID
- IP address

Duplicated controller names

If the Alias reference to the robot controller cannot be resolved, than the application will not be able to communicate with the robot. You must define the type of association for an Alias carefully, as various associations behave differently.

Example

IRC5 OPC UA Server uses an Alias to reference a specific robot controller. Alias can only use one Controller Name to uniquely identify a robot controller, to ensure that every robot controller on your network has a unique combination of parameter settings. Duplicated Controller Names and parameters will cause ambiguousness, and the IRC5 OPC UA Server can no longer distinguish between the duplicated names.

Note

Duplicated controller names causes problem to the network, as the Alias resolves to the controller which is online at the moment. If one controller is disconnected and a second one brought online using the same parameter settings, the Alias will resolve to the new controller. However, only the OPC UA items already existing in the namespace will be updated with the new values. The new variables will not be added to the namespace unless the OPC UA Server is restarted.

Continues on next page

2.2.2 Aliases
Continued

Address

The following table provides information on IP address parameters for alias:

If...	Then...
you associate an Alias with only the parameter IP address	you should make sure that the IP address is statically assigned to the robot controller.
you use DHCP	your DHCP server must be carefully configured in order to guarantee repeatable assignments of the IP address.
the IP addresses are reassigned	the Alias might resolve to a different controller.

System name

The following table provides information on system name parameters for alias:

If...	Then...
you associate an Alias with only the parameter System Name	you must ensure that the System Name is unique for each robot controller. This can be inconvenient if you have several robots that could otherwise be loaded with the same RobotWare system (and thus be given the same System Name). The same resolution difficulties exist as in the case of duplicate Controller Names.

Controller ID

The following table provides information on controller ID parameters for alias:

If...	Then...
you associate an Alias with only the parameter Controller ID	you can be sure that the Controller ID is unique. A Controller ID is permanently assigned to a specific robot controller.
you replace the robot controller hardware	the Controller ID will change, and you must then redefine the Alias to associate it with the new controller.

System ID

The following table provides information on system ID parameters for alias:

If...	Then...
you associate an Alias with only the parameter System ID, and you are certain that the System ID is unique	you must remember that the System ID will change when you load next time a RobotWare system to the robot controller. If you need to make any modifications to the RobotWare system configuration that would result in a subsequent reload, the System ID will change and you must redefine the Alias.

Recommended associations

ABB recommends that you define Aliases to ensure stability of the association of the Alias to a particular robot.

If you use static IP addressing or your DHCP server which is configured in such a way that its IP address assignment is repeatable, you should associate Alias definitions with both Controller Name and IP address. Using this approach, you can download new RobotWare systems, or replace an entire controller without the need to redefine the Alias.

Continues on next page

If you cannot guarantee stable IP addresses, then you should define Aliases using both the Controller Name and Controller ID Connection Criteria. With this approach, you can change the IP address, or you can download new BaseWare systems without the need to redefine the Alias. However, if you replace controller hardware, the Controller ID will change and you must redefine the Alias.

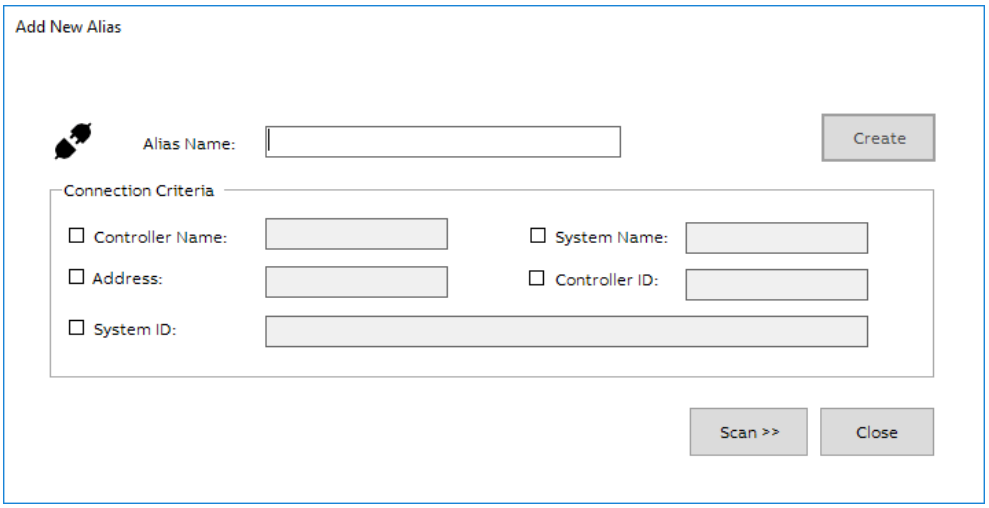
Either approach reliably ensures that an Alias will always resolve to a unique, well-known robot controller.

2.2.3 Add New Alias dialog screen components

Add New Alias dialog components

In the main screen click on Add Alias icon.

The **Add New Alias** window is displayed.



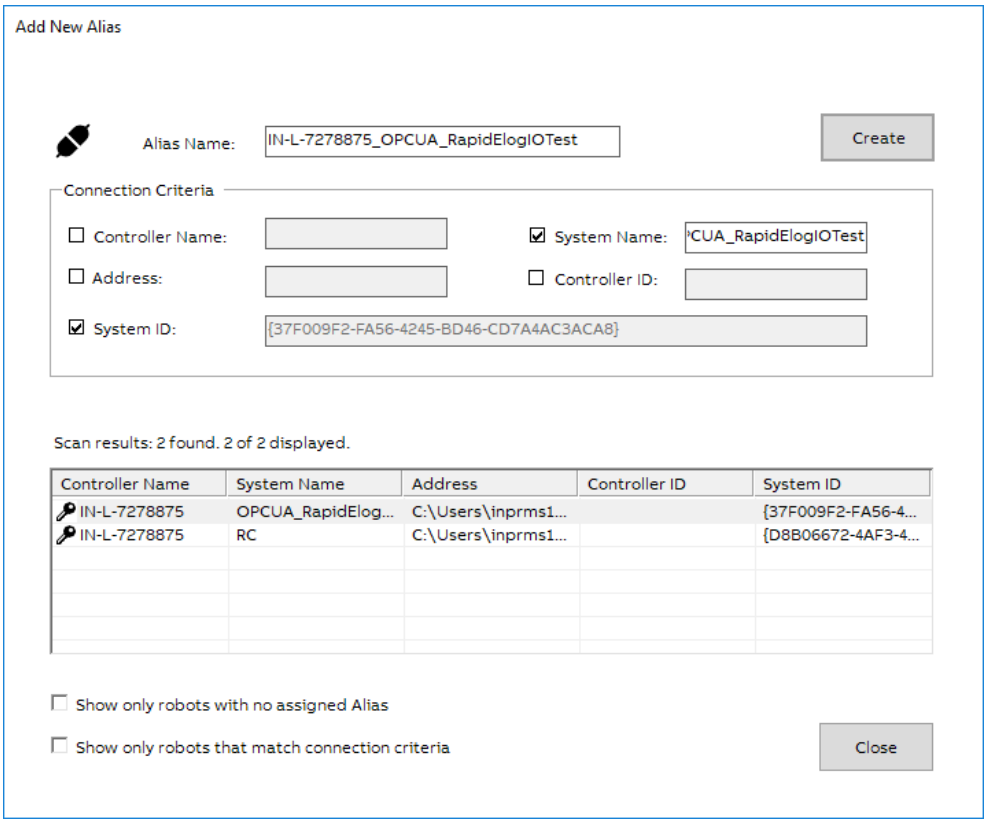
xx2000000019

Component	Description
Alias Name field	This field allows you to enter the Alias name.
Create button	Click to create a new Alias. This button becomes active after selecting any connection criterion.
Connection Criteria check boxes	• Controller Name - This is the name assigned to the robot controller. • System Name - This is the name assigned to the currently active BaseWare system in the robot controller. • Address - This is the robot controller's IP address. If the controller is a Virtual Controller (VC), this is the path to the VC system directory. • Controller ID- -This is the robot controller's unique hardware ID. • System ID - This is the ID assigned to the currently active BaseWare system in the robot controller.
Scan button	Displays the list of available controllers.

Continues on next page

Scan Feature screen components

In **Add New Alias** window, click on **Scan** button to display the scan information.



xx2000000036

Component	Description
Scan Results list box	Shows a list of all of the IRC5 robots detected on the network.
Show ...Alias check box	Click to show only those robots that do not have an Alias assigned.
Show ...criteria check box	Click to show only those robots whose criteria match the criteria selected in the Connection Criteria fields.
Alert Message	Appears any time one or more robots match the connection criteria selected.

Add New Alias dialog icons

The following icons can be displayed in the Controller Name list in the Scan Feature dialog.

Icon	Description
 Unaccessable Alias	The alias is connected to the network, but the PC Interface and/or the OPC UA Server RobotWare option is missing.

Continues on next page

2.2.3 Add New Alias dialog screen components
Continued

Icon	Description
 Field locked	When one of the fields is modified manually, a small lock will appear next to it. This indicates that from now on, the configuration tool will not modify these fields automatically, but let the user enter the the values manually.
 Accessable Alias	The alias is connected to the network, and the PC Interface and/or the OPC UA Server RobotWare option is present.

2.2.4 Edit Alias dialog components

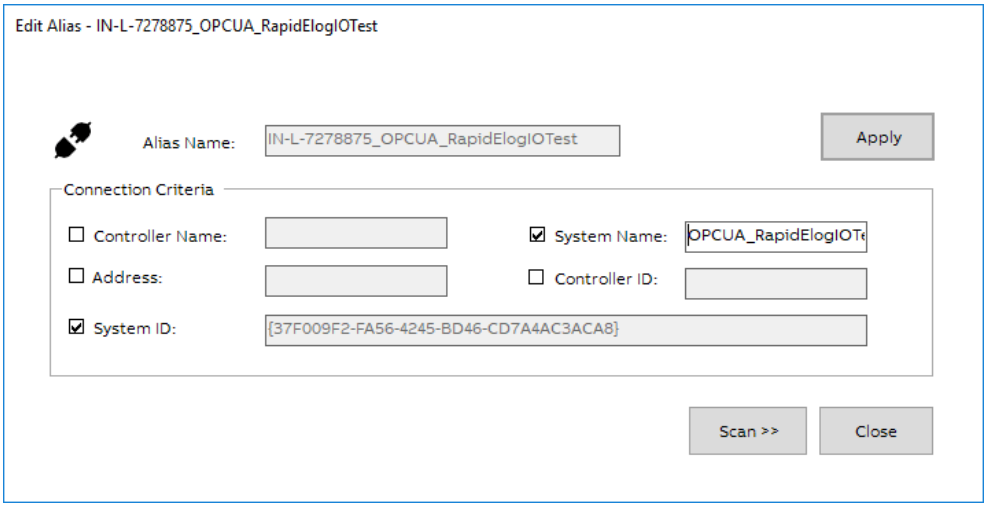
Overview

This section describes components displayed on **Edit Alias** window.

Edit Alias dialog

In the main screen select the alias from the list under **Alias** tab and click on the **Edit Alias** icon.

The **Edit Alias** window of the selected alias is displayed.



xx2000000035

Component	Function
Alias Name field	This field is disabled by default.
Connection Criteria check boxes	Criteria: • Controller Name - This is the name assigned to the robot controller. • System Name - This is the name assigned to the currently active RobotWare system in the robot controller. • Address - This is the robot controller's IP address. If the controller is a Virtual Controller (VC), this is the path to the VC system directory. • Controller ID -This is the robot controller's unique hardware ID. • System ID - This is the ID assigned to the currently active RobotWare system in the robot controller.
Scan button	Displays the **Scan Results** list box.
Close button	Close the Display.

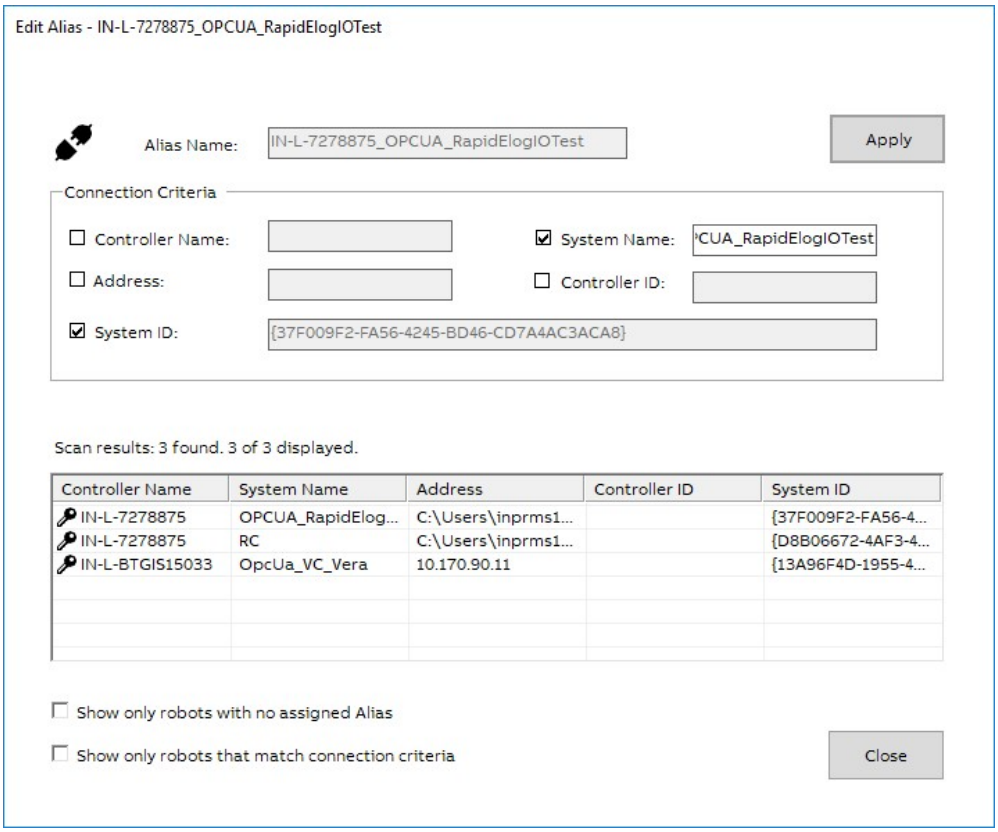
Continues on next page

2.2.4 Edit Alias dialog components

Continued

Scan Feature Screen Components

In **Edit Alias** window, click on **Scan** button to display the scan information.



xx2000000183

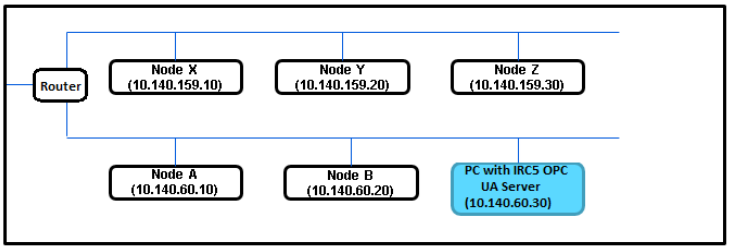
Component	Function
Scan Results list box	Shows a list of all of the IRC5 robots detected on the network.
Show only robots with no assigned Alias check box	Click to show only those robots that do not have an Alias assigned.
Show only robots that match connection criteria check box	Click to show only those robots whose criteria match the criteria selected in the Connection Criteria fields.

2.2.5 Add remote controller

Adding remote controller

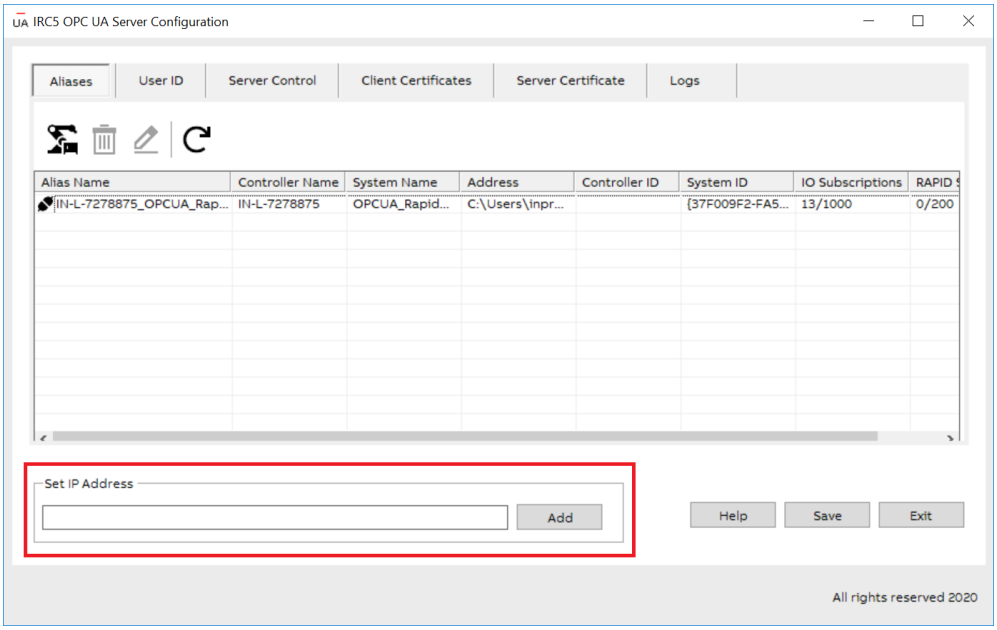
You can add a remote controller from multiple subnets.

The following image is a representation of remote controllers in different subnets:



xx1500000086

Node	Description
A,B	Controllers in same subnet.
X,Y,Z	Controllers in another subnet.



xx2000000176

Follow the below steps on how to add a remote controller:

1. Type the IP address of the remote controller in the **Set IP Address** text box.
2. Click **Add**.

   The controller is added to the IRC5 OPC UA Server.

2.2.6 The OPC UA Alias configuration file

Overview

As discussed in the *Getting started on page 11*, each IRC5 robot you wish to communicate to the ABB IRC5 OPC UA Server must contain an Alias definition in the OPC Alias Configuration file.

During installation of the ABB IRC5 OPC UA Server, the installation program places a file called OPCUAConfig.xml in a subdirectory of this location, “ C:\ProgramData\ABB\IRC5 OPC UA”.

OPCUAConfig.xml

```
<ABB_IRC5_OPC_Configuration version="1">
    <UAS Username="Default User" Password="robotics" />
    <OPCServerLanguage Language="en" />
    <AliasList>
      <Alias Name="GRUMPY" PCI="Connected">
        <Address IP="192.168.8.105" CTRLID="" SID="" NAME=""
            SYSNAME="RW5_05_032" />
      </Alias>
      <Alias Name="Duck" PCI="Connected">
        <Address IP="" CTRLID="" SID="" NAME="" SYSNAME="ROL_505_31"
            />
      </Alias>
      <Alias Name="Elvis" PCI="NoPCI">
        <Address IP="192.168.8.111" CTRLID="" SID="" NAME=""
            SYSNAME="eio__test" />
      </Alias>
    </AliasList>
  </ABB_IRC5_OPC_Configuration>
```

UAS tag stands for User Authorization System. It stores the username and the password used to log on to the robot controller.

Alias definition

Each alias definition within this file consists of the following three lines of XML syntax:

```
<Alias Name="Elvis" PCI="NoPCI">
  <Address IP="192.168.8.111" CTRLID="" SID="" NAME=""
      SYSNAME="eio__test" />
</Alias>
```

PCI value

Each alias definition has a PCI value, which specifies one of the following:

- Connected - the specified robot controller is connected and has the "616-1 PC Interface” and/or the "1582-1 OPC UA Server" RobotWare option installed.
- Disconnected - either the alias cannot be resolved to a single robot controller on the network, or there is no such robot controller connected to the network.
- NoPCI - PC Interface RobotWare option is not installed.

Continues on next page

Parameters

Each alias definition consists of five parameters, which are specified as XML element attributes; these parameters correspond to:

- The alias name is the Name attribute of the Alias element; e.g. “ELVIS”. This is the name you want the robot to be identified as by the OPC Server.
- The robot’s IP Address (the IP attribute of the Address element); e.g. “130.110.69.254”
- The robot’s Controller ID (the CTRLID attribute of the Address element).
- The robot’s SystemID (the SID attribute of the Address element).
- The robot’s Controller Name (the NAME attribute of the Address element); e.g. “GRUMPY”
- The robot’s System Name (the SYSNAME attribute of the Address element).

Note

Not all parameters necessarily need to contain values, according the association rules described in the section *Aliases on page 21*.

2.2.7 OPC UA Server username/password

Enter the username

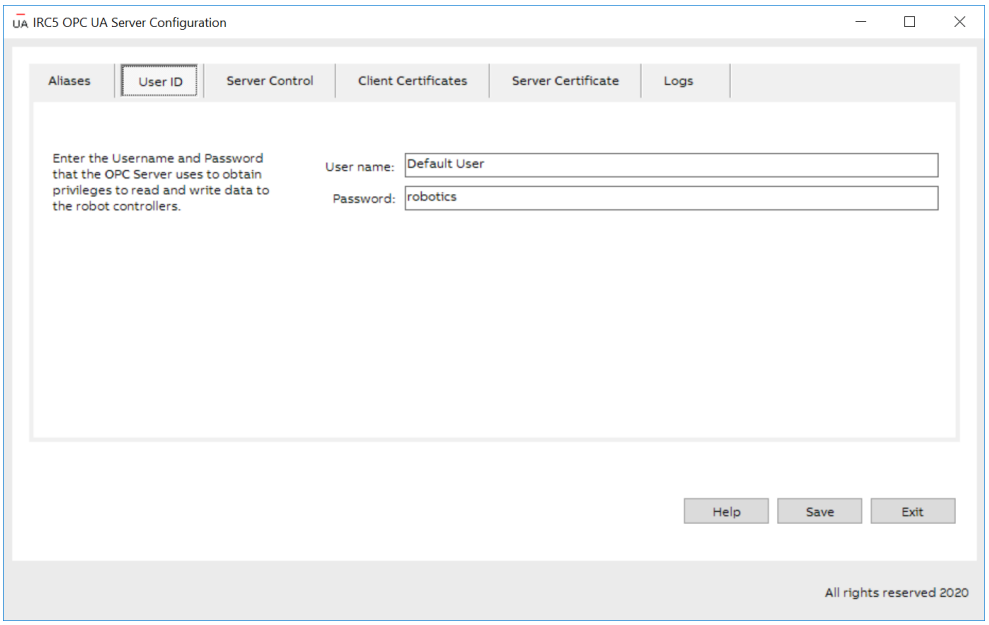
Enter the username and password in ABB IRC5 OPC UA Server to obtain access to read and write data of the robot controllers.

> **Note**
>
> The ABB IRC5 OPC UA Server uses the same username and password combination for each and every robot controller that is accessed.

> **Note**
>
> Empty username is not allowed.

Username settings

In the main screen click on User ID tab.



xx2000000020

To set the user settings:

Step	Action	Information
1	Enter a Username.	
2	Enter a Password.	
3	Click **Save** to save the information entered.	

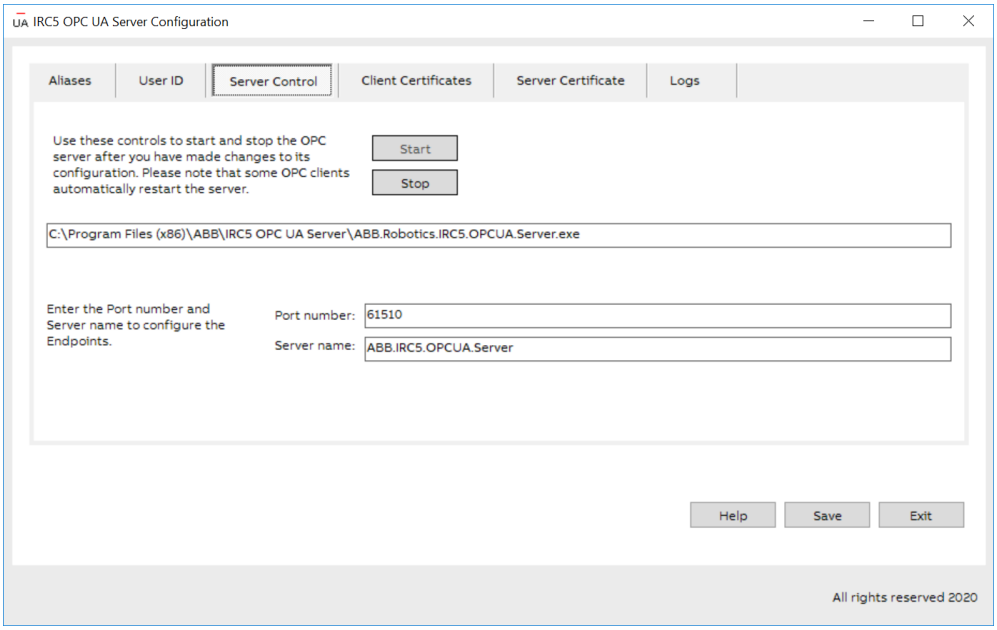
2.2.8 Server Control

Overview

In this section you can control start and stop of the OPC server after you have made changes to its configuration.

Note

You can also automatically restart the server using this option.

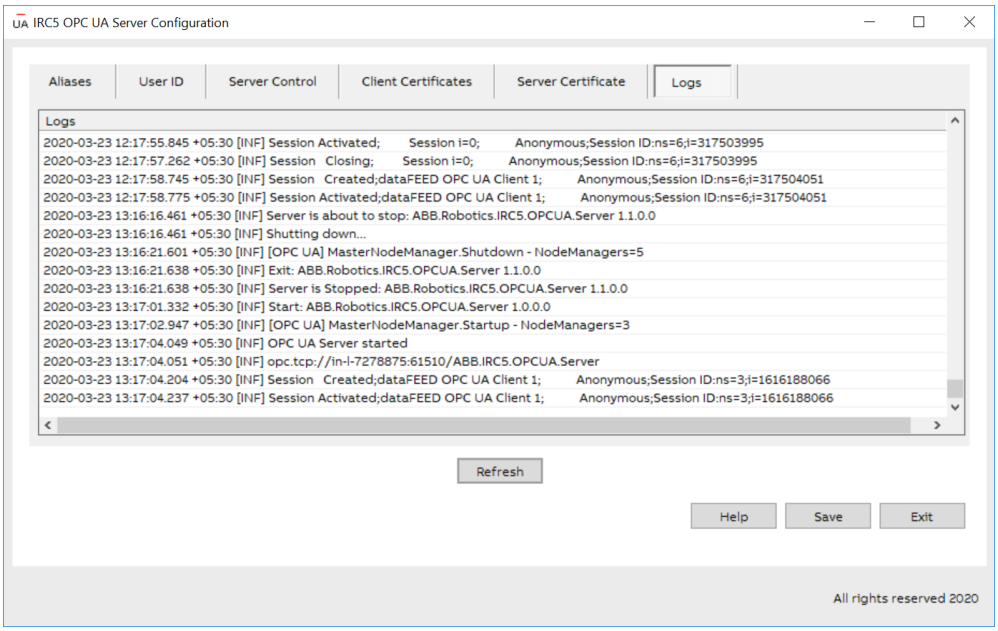


xx2000000254

2.2.9 Logs

Overview

In this all the log informations on the operation done are listed.



xx2000000257

- Latest logs can be seen in **Logs** tab in OPC UA Configuration tool.
- IRC5 OPC UA Server log files can be found at location *C:\ProgramData\ABB\IRC5 OPC UA*.
- IRC5 OPC UA Server configured to store only latest 10 log files.

2.3 How to add/edit IRC5 robot alias

2.3.1 How to add new robot aliases

Entering information manually

In this procedure, you enter an Alias name and the Connection Criteria necessary to reliably identify the controller for which you are creating the Alias.

Tip

See ABB's Recommended Associations in section *Aliases on page 21* to reliably identify controllers.

Adding new alias

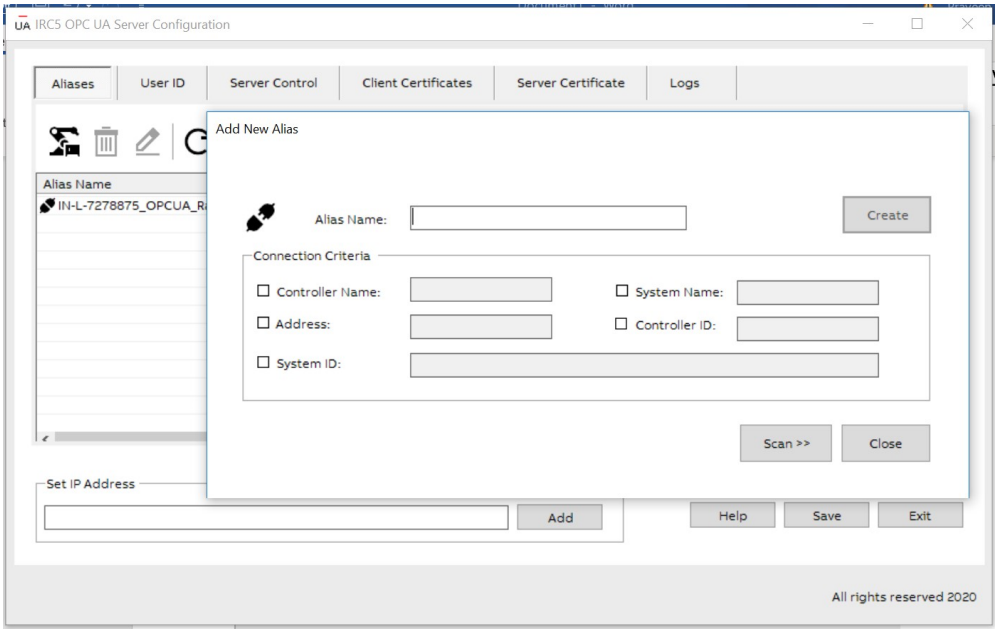
Follow the steps below on how to add a new IRC5 Robot Alias to the ABB IRC5 OPC UA Sever Configuration application in main screen.

Step	Action	Information
1	Click on the **Add New Alias** icon  from the main screen.	The **Add New Alias** window is displayed.
2	Enter an Alias name in the **Alias Name** field.	
3	Enter the robot's controller name in the **Controller Name** field.	
4	Enter the robot's controller IP address in the **Address** field.	
5	If required enter information for (System ID, Controller ID, etc.) to further identify the controller for which you are creating the Alias.	
6	Click **Create**. The Alias you created now appears on the Aliases tab.	

Continues on next page

2.3.1 How to add new robot aliases

Continued



xx2000000175

Using the Scan Feature

Follow the steps below to use the IRC5 Robot Scan feature to detect IRC5 robots connected to the network:

> **Note**
>
> For information on scan feature components, refer *Scan Feature screen components on page 25* section.

Step	Action	Information/Illustration
1	Click on the **Add New Alias** icon from the main screen.	The **Add New Alias** window is displayed.
2	Click **Scan**.	The **Scan Results** list box will appear, showing a list of robots detected on the network. You can filter the list as shown.
3	Select the controller that you wish to create an Alias for from the list. The ABB IRC5 OPC UA Server Configuration application displays a default name for the Alias in the **Alias Name** field, based on the Controller Name and System Name.	
4	Select the Connection Criteria that you wish to use to identify the robot. You can select more than one criterion.	**Note** **Note:** See ABB's Recommended Associations to reliably identify controllers in *Aliases on page 21*.
5	If necessary, you can change the Alias Name as required.	

Continues on next page

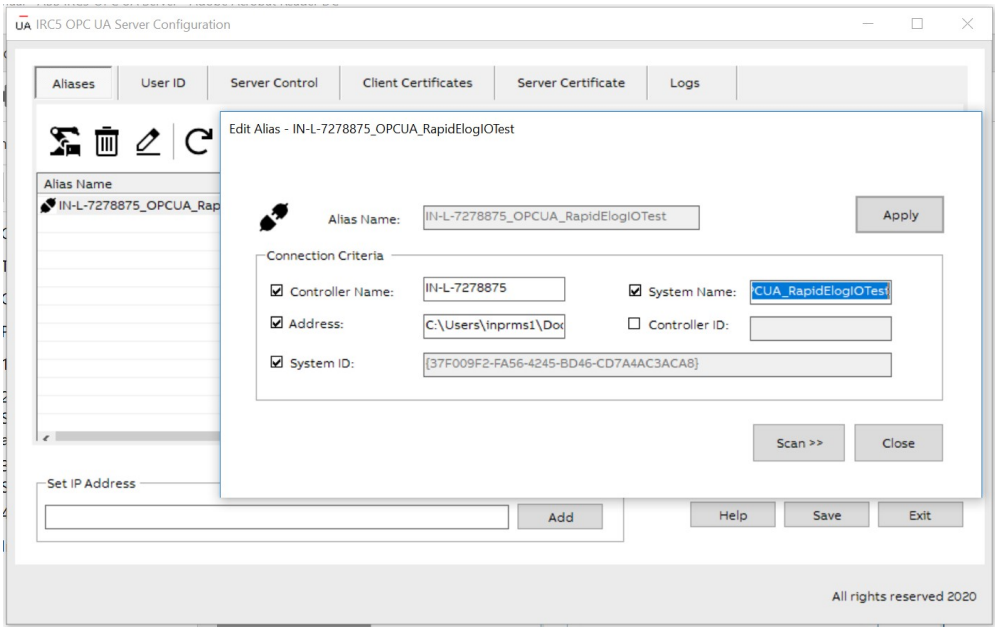
Step	Action	Information/Illustration
6	Click **Create**. The Alias you created now appears on the ABB IRC5 UA OPC Server Configuration application main screen.	

2.3.2 How to edit an IRC5 robot alias

Editing an alias

Follow the steps below to change the association parameters for the selected robot Alias.

Step	Action	Information
1	Click on the **Edit Alias** icon from the main screen.	The **Edit Alias** dialog will appear. **Note:** That the Alias name field is not editable.
2	Click on one or more **Connection Criteria** check boxes and enter the required information.	
3	Click **Apply**.	



xx2000000179

Using the Scan Feature

Follow the steps below to use the IRC5 Robot Scan feature to detect IRC5 robots connected to the network.

> **Note**
>
> For information on scan feature components, refer *Scan Feature Screen Components on page 28* section.

Step	Action	Information/Illustration
1	Click on the **Edit Alias** icon from the main screen.	The **Edit Alias** dialog will appear.

Continues on next page

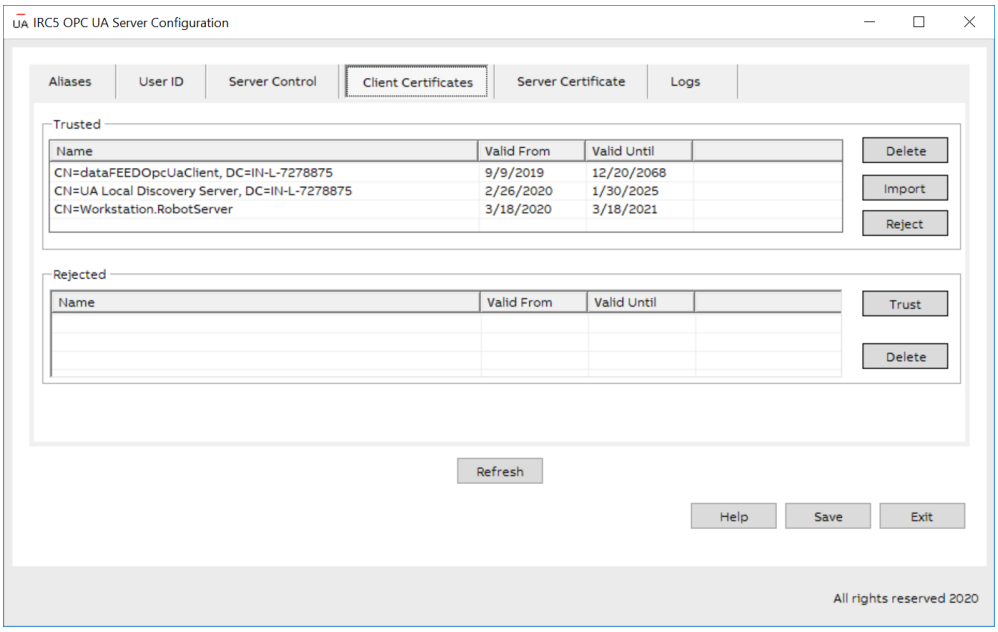
Step	Action	Information/Illustration
2	Click **Scan**.	The **Scan Results** list box will appear, showing a list of robots detected on the network. You can filter the list as shown here.
3	Select the controller that you wish to create an Alias for from the list. The ABB IRC5 OPC UA Server Configuration application displays a default name for the Alias in the **Alias Name** field, based on the Controller Name and System Name.	
4	Select the Connection Criteria that you wish to use to identify the robot. You may select more than one criterion.	**Note** **Note:** See ABB's Recommended Associations to reliably identify controllers in *Aliases on page 21*.
5	If necessary, you can change the Alias Name as required.	
6	Click **Apply**. The edited Alias appears on the ABB IRC5 OPC UA Server Configuration application main screen.	

2.4 Certificate management

2.4.1 Client Certificates

Overview

The Client Certificates tab provides an interface for managing the configuration of the security certificates for the application. It allows the user view trusted client certificate list and rejected certificate list.



xx2000000151

Trusted client certificates list

Displays the list of certificate that the application trusts. This list includes any Certificate Authority (CA) certificates. Administrator can use this list to check the expiration dates of certificates and renew any certificate prior to their expiration. The administrator can also use the list to ensure that only application that are authorized applications are in the trust list.

- Delete
- Import
- Reject

Delete Trusted Certificate(s)

Delete option allows an administrator to delete the certificate from trusted list.

Import Certificate(s)

Import option allows an administrator to select a certificate and add it to the list of trusted certificates. The certificate can be on any location including flash drives or network share locations. It is the administrator's responsibility to review the certificate and ensure that it is a certificate that belongs to an application that is to be trusted.

Continues on next page

Import option allows an administrator to import an entire list of certificates or to move a list from one store to another. This feature can be used along with a network share to build a list of certificates that are to be stored and then import the list for a new installation.

Reject client certificate(s)

Reject option allows an administrator to reject the certificate(s) from trusted list.so that rejected certificate(s) shall be moved from trusted list to rejected list.

Rejected client certificates list

A Rejected list of Certificates shall contain certificates which are rejected by IRC5 OPC UA Config tool (or) which are rejected by administrator.

- Trust
- Delete

Trust the rejected client certificate(s)

Trust option allows an administrator to trust an entire list of certificates (or) to trust a client certificate. Which will move the client certificate(s) from rejected list(s) to trusted list.

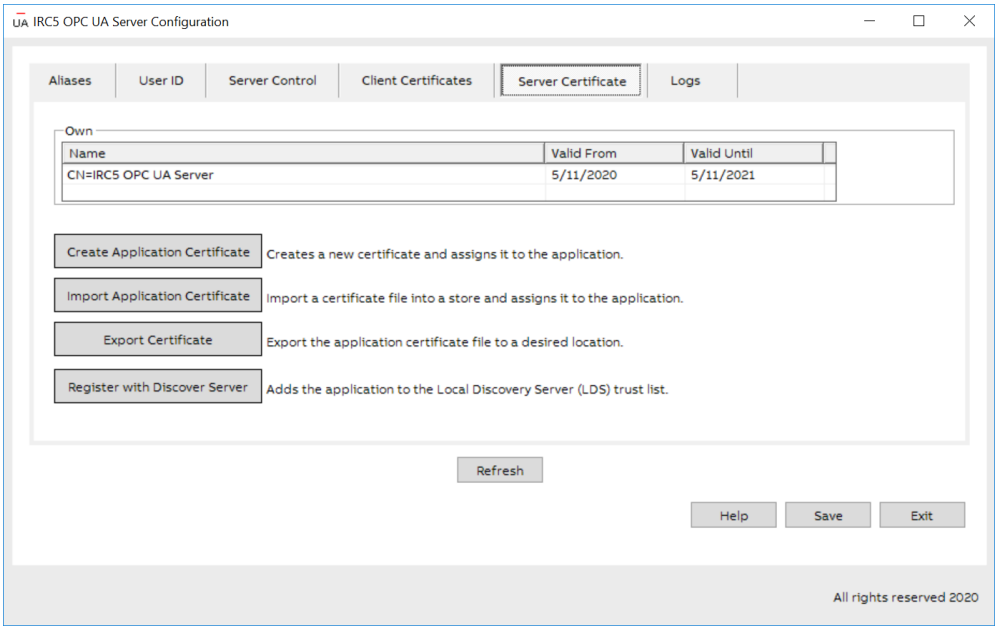
Delete Rejected Certificate(s)

Delete option allows an administrator to delete the certificate(s) from rejected list.

2.4.2 Server Application instance Certificates

Overview

In section you can add the OPC UA server to Local Discovery Server (LDS) to trust list and also you can create a new certificate and assign it to the application.



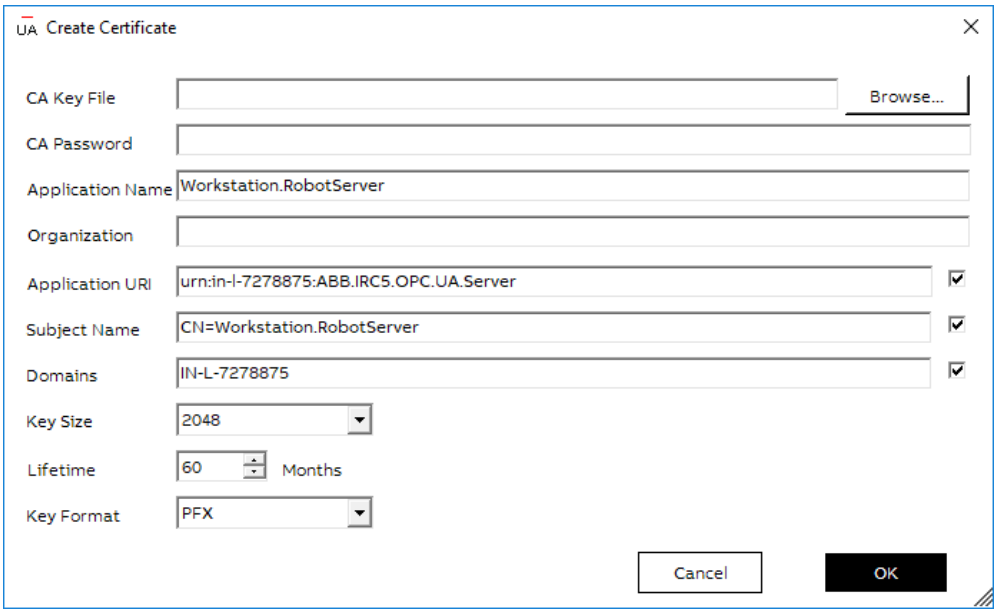
xx2000000255

Options	Description
Create Application Certificate	Creates a new certificate and assigns it to the application.
Import Application Certificate	Import a certificate file into a store and assign it to the application.
Export Certificate	Export application certificate file to the desired location.
Register with Discover Server	Adds the application to the Local Discover Server (LDS) trust list.

Continues on next page

Create Application Certificate

This option creates a new application certificate with the following dialog.



xx2000000256

Options	Description
Store Path	Specifies where the application certificate will be placed after it is created.
CA Key File	The CA Key File is a .pfx file containing the Certificate Authority private key. If left blank a self-signed certificate is created. If provided then the CA Password is required.
CA Password	Displays the password required for CA Key File.
Application Name	Displays the name of the application.
Organization	Displays the name of the organization.
Application URL	Displays the name of the application URL. **Note** If checkbox is unchecked then the tool will generate information automatically from the other information provided
Subject Name	Displays the subject name. If checkbox is unchecked then the tool will generate information automatically from the other information provided
Domains	Displays the name of the domain. If checkbox is unchecked then the tool will generate information automatically from the other information provided
Key Size	Displays the size of the key.
Lifetime	Displays the validity of the key.
Key Format	Displays the format of the key.

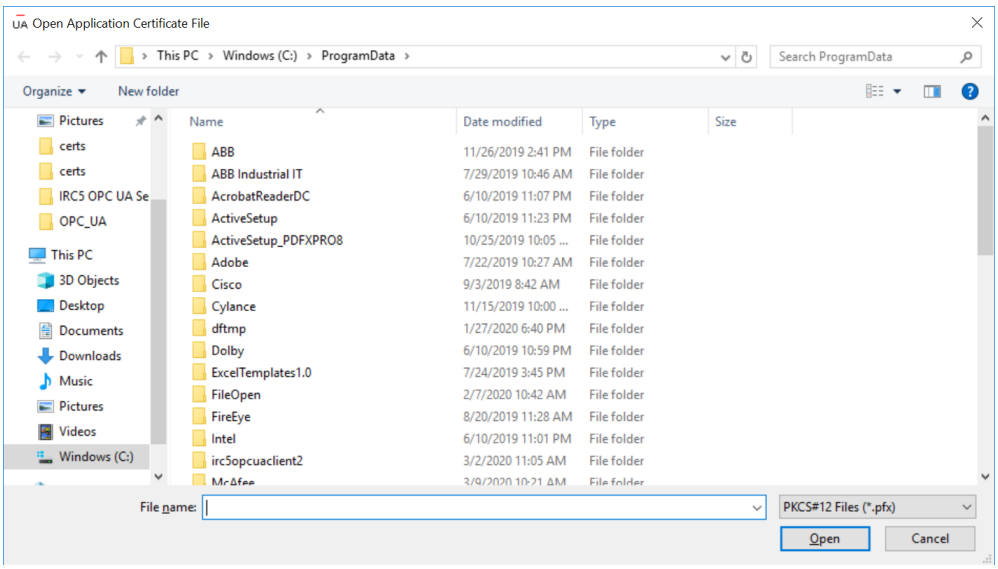
Continues on next page

2.4.2 Server Application instance Certificates

Continued

Import Application Certificate

This option imports an application certificate from a .PFX file stored on disk. It prompts the user to enter a password if one is required. The imported Certificate will replace any Application Instance certificate that may already be assigned to the application.



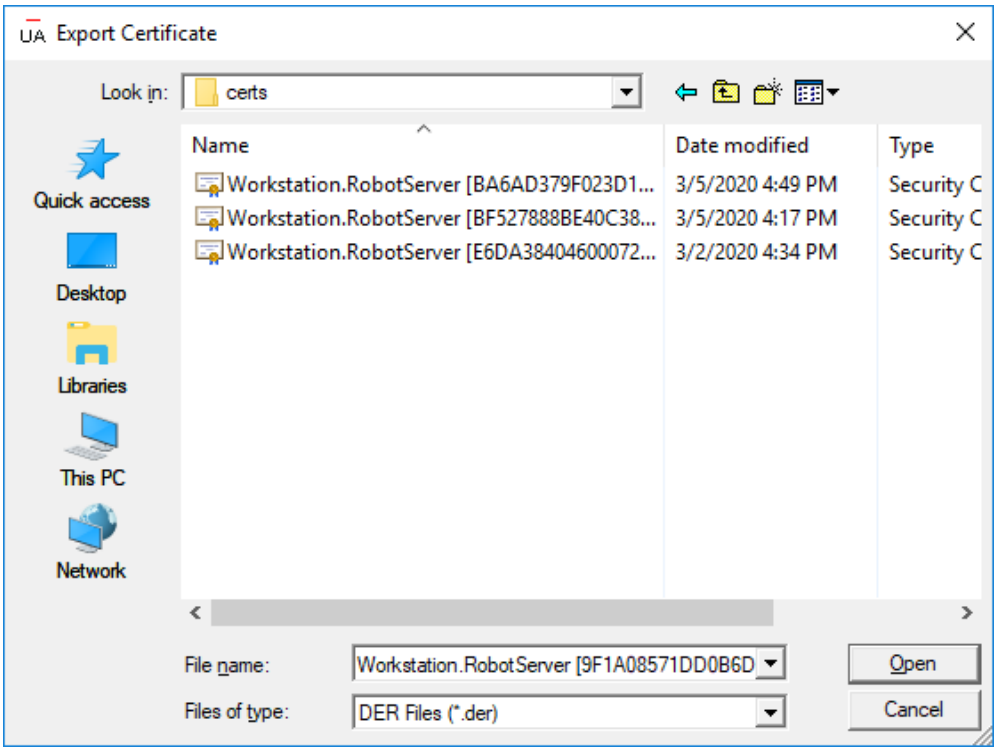
xx2000000294

Application instance certificate can be created by open source tools like using *UA Configuration Tool* and export the certificate using IRC5 OPC UA Configuration tool.

Continues on next page

Export Certificate

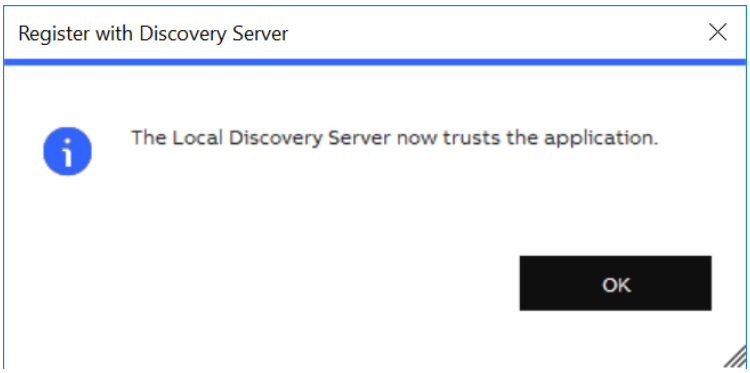
This option allows an administrator to generate a file that contains the certificate. This file can then be used to import the certificate onto another machine. The resulting file is a .DER file and does not contain the Private Key information.



xx2000000292

Register with Discover Server

This option allows an administrator to register the selected application's certificate with the Discovery server as a trusted certificate.



xx2000000379

OPC UA Config tool will give the warning message **The Local Discovery Server is not installed** when Local Discovery Server (LDS) is not installed in the system.

For more details on **Local Discovery Server**, please see the link:

https://opcfoundation.org/developer-tools/samples-and-tools-unified-architecture/local-discovery-server-lds/

This page is intentionally left blank

3 ABB IRC5 OPC UA Server

3.1 Address space

Introduction

The OPC UA Data Access function of the ABB IRC5 OPC UA Server is to read and write data managed by the ABB IRC5 robot controller.

Data items in ABB IRC5 OPC UA Server are referred by their node names.

The ABB IRC5 OPC UA Server presents various predefined nodes that provide information concerning to the robot controller's current state. In addition to these predefined nodes, the OPC UA Server presents up to 1000 additional nodes that contain the values of the IRC5's I/O signals, as well as up to 200 nodes that contain the values of the IRC5's RAPID data values for each IRC5 controller configured in the OPC UA Server..

IRC5 Information Model

The IRC5 Information Model is an ABB Robotics Proprietary OPC UA Information Model for robot controllers.

The tags exposed by the ABB IRC5 OPC UA Server follow the hierarchical structure of the IRC5 Controller object model.

The following tags in the Controller domain in the IRC5 Controller object model diagram are:

- All of the RAPID and IOSYSTEM tags
- OperatingMode
- ControllerState
- ControllerExecutionState
- SpeedRatio
- MasterRAPID
- MasterCFG
- InterfaceState

The following unsupported tags in the Controller domain in the IRC5 Controller object model diagram are:

- CollisionDetectState
- RapidProgramFreememory
- RapidProgramUsedMemory

The following tag is updated when OPC client application requests an update from the server:

- SystemClock

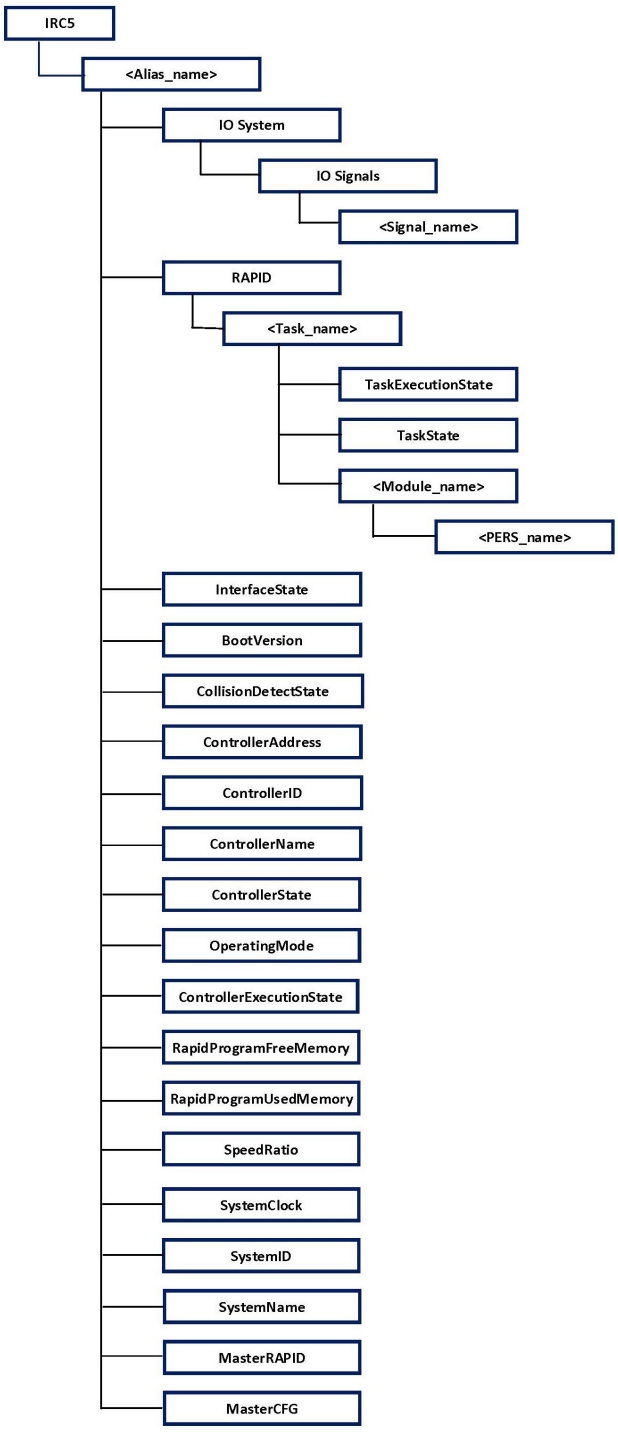
All Other items are updated only at the controller restart.

Continues on next page

3.1 Address space

Continued

The IRC5's Objects and Variables can be seen in the image below:



xx2000000068

Note

For detailed description of the OPC UA Information Model for IRC5 robot controllers, refer *Appendix A - ABB Robotics OPC UA proprietary information model for IRC5 on page 63*.

Continues on next page

OPC UA Robotics Companion Specification

The OPC UA Companion Specification for Robotics specifies an OPC UA Information Model for the representation of a complete motion device system as an interface for higher-level control and evaluation systems. A motion device system consists out of one or more motion devices, which can be any existing or future robot type (e.g. industrial robots, mobile robots), kinematics or manipulator as well as their control units and other peripheral components.

The IRC5 OPC UA Server supports the mandatory parts and some of the optional parts of the OPC UA Companion Specification for Robotics.

Please refer *OPC 40010-1 - Robotics Part 1: Vertical Integration*.

Subscription for Data Changes

A OPC UA Client can subscribe to multiple Nodes, which are monitored by OPC UA Server. OPC UA server notifies the OPC UA client about value changes.

IRC5 OPC UA Server supports subscriptions for data changes of variable values for below mentioned nodes:

- OperatingMode
- ControllerState
- ControllerExecutionState
- MasterRAPID
- MasterCFG
- IO-Signal value changes
- RAPID persistent variable value changes

The maximum number of subscriptions for one session is 1000.

3.2 Events implementation

Overview

An event monitored item is a special type of a monitored item designed for receiving event notifications from the UA Server. For creating this kind of object, the user needs to create a monitored item for the EventNotifier attribute of an object node. The object node needs to have the SubscribeToEvents bit mask set in the EventNotifier attribute in order to allow the creation of event monitored items.

Event log event

All IRC5 OPC UA event log are OPC generated Events. Some of the parameters included in the event structure that may have special meaning in the IRC5 context are described below. (This is not a comprehensive list. Please refer to OPC UA Alarms and Events specification for a complete list.)

- **Source–** The alias name of the controller that generated the event.
- **Message–** The title or brief explanation of the event.
- **Event Category –** The IRC5 Event Log category.
- **Severity–** The severity of the event.

The following parameters contain event data if attribute values are requested by the client.

- **Number of Event Attributes–** The length of the event attribute array.
- **Event Attributes –** A pointer to the ABB specific event attributes as requested by the client according to the OPC specification.

Source Name	Time	Message	Severity
IN-L-BTGIS15033_OpcUa_VC_Vera	10:24:17.000 AM	Automatic mode confirmed	100
IN-L-BTGIS15033_OpcUa_VC_Vera	10:24:16.000 AM	Motors OFF state	100
IN-L-BTGIS15033_OpcUa_VC_Vera	10:24:15.000 AM	Speed adjusted	100
IN-L-BTGIS15033_OpcUa_VC_Vera	10:24:15.000 AM	Automatic mode requested	100
IN-L-BTGIS15033_OpcUa_VC_Vera	10:24:15.000 AM	Manual mode selected	100
IN-L-BTGIS15033_OpcUa_VC_Vera	10:24:15.000 AM	Safety guard stop state	100

xx2000000178

Category ID

The IRC5 Event Log categories are shown in the following table.

Category ID	Description
0	Common
1	Operational
2	System
3	Hardware
4	Program
5	Motion
7	IO

Continues on next page

Category ID	Description
8	User Defined
9	Safety
10	Internal
11	Process
12	Configuration
15	RAPID

Event severity level

The IRC5 OPC Alarms & Events Server automatically translates ABB IRC5 event log types to specific severity levels as shown in the table below.

Event log type	Severity
State Change	100
Warning	300
Error	600

Event log attributes

In addition to the standard attributes required by the OPC Alarms and Events specification, the IRC5 OPC Alarms & Events Server can provide the attributes defined in the table below.

Attribute	Attribute ID
Number	1
Description	2
Action	3

Summary

- **Number–** The message number is calculated by multiplying 10000 by the category ID and then adding the event message ID. For example: If we have a category ID of 1 (operational events) and the message ID is 51, the message number is 1 * 10000 + 51 =10051.
- **Description–** A detailed description of the event. This attribute will be empty if there is no description for a specific event log.
- **Action** – Description of actions related to the event. This attribute will be empty if there is no description for a specific event log.

3.3 Security

3.3.1 Transport Protocols

Overview

The OPC UA specification currently defines two data encoding, multiple security protocols and two transport protocols.

Data Encodings

Data encodings are available for XML and UA Binary. IRC5 OPC UA Server uses UA Binary encoding.

UA Binary: This message format encodes the data serialized into a byte array. UA Binary offers reduced computational cost in terms of encoding and decoding but can only be interpreted by OPC-UA compliant clients. UA Binary is more likely to be used in device level communications where processing power is limited and performance is a high priority.

Security Protocols

A security protocol ensures the integrity and privacy of UA messages that are exchanged between OPC UA applications.

There are two security protocols defined for OPC UA:

- WS Secure Conversation
- UA Secure Conversation

IRC5 OPC UA Server supports UA Secure Conversation.

Transport Protocols

OPC UA supports the following transport protocols:

- OPC UA TCP
- SOAP/HTTP
- HTTPS

IRC5 OPC UA Server supports OPC UA TCP transport protocol.

OPC UA TCP: This is a TCP (sockets) based protocol providing a full duplex channel between client and server. Messages are packaged into a structure specified by the OPC UA TCP binary protocol and the structure is transmitted using a socket or secure socket (depending on the endpoints security requirements). As OPC UA TCP is specific to the OPC UA specification only OPC UA Clients and Servers are capable of receiving data transmitted with OPC UA TCP.

3.3.2 Security configuration

Overview

When securing the communication with the OPC UA protocol, the following settings are required:

- Security Policies
- User Token Policies

Security Policies

Security policy and SecurityMode (message mode) parameters specify the security algorithms that the UA server supports.

- **Security policy**:

  Selection of cryptographic algorithms. Any existing client and server which needs to interact should support this policy. Weaker security policies use outdated algorithms and should not be used. At a minimum, the Security Policy '**Basic256Sha256**' should be chosen.

  IRC5 OPC UA Server uses following security policies:

  - Basic256Sha256
  - Aes128_Sha256_RsaOaep
  - Aes256_Sha256_RsaPss

- **SecurityMode:**

  The SecurityMode should be 'Sign' or 'SignAndEncrypt'. This ensures that, authentication at the application level is enforced. The SecurityMode 'None' does not provide any protection. SecurityMode 'SignAndEncrypt' provides integrity and confidentiality for the data.

  OPC UA supports the following security modes:

  - **None** - no encryption, security is turned off. Messages can be read by a 3rd party and tampered with.
  - **Sign** - messages are signed to ensure data integrity but the message body is unencrypted. Messages can be read by a 3rd party.
  - **Sign and encrypt** - as above but with the message body encrypted. Secure, messages are private and their integrity is assured.

  IRC5 OPC UA Server by default supports only 'Sign' and 'Sign and encrypt'. 'None' security mode is disabled.

> **Note**
>
> 'None' security mode can be enabled in IRC5 OPC UA Server configuration file. But it is not recommended for normal use due to risk of unintentional or malicious activity.

Continues on next page

3.3.2 Security configuration
Continued

> **Note**
>
> Default security policies in OPC UA server config xml can only be changed by an expert (administrator).

User Token Policies

OPC UA Applications supports authentication of users by providing the necessary authentication credentials to the other entities. OPC UA Applications accept tokens in any of the following forms:

- **Anonymous:** No user information is available
- **UserName:** A user identified by user name and password
- **X509v3:** A user identified by an X509v3 Certificate
- **WSS:** A user identified by a WS-SecurityToken. (e.g. SAML,Kerberos-Ticket)

IRC5 OPC UA Server supports only Anonymous, UserName user token policies.

> **CAUTION**
>
> The identifier ‘anonymous’ should be used “only for accessing non-critical UA server resources as it does not provide any protection. It is not possible to trace who has changed the data or configuration on the server side when this generic identifier is used. Also, an attacker could use this identifier to read or write data in an unauthorized manner if no adequate restriction of the rights of the identifier ‘anonymous’ was configured

Server Application Instance Certificate(X.509 certificate)

When the UA application starts, it attempts to locate and retrieve the certificate configured in the <ApplicationCertificate> section of the application configuration.

In case the OPC UA stack cannot find the certificate specified by the configuration, it will attempt to create a self-signed certificate using some internal methods. The newly created certificate will be saved at the location specified by configuration in order to be found at the next run of the IRC5 OPC UA Server. Authentication of clients and servers achieved by using application instance certificates (X.509 certificates). Please refer *Create application certificate* and *Import application certificate* using IRC5 OPC UA Configuration tool.

3.3.3 Restricting Access to application folder

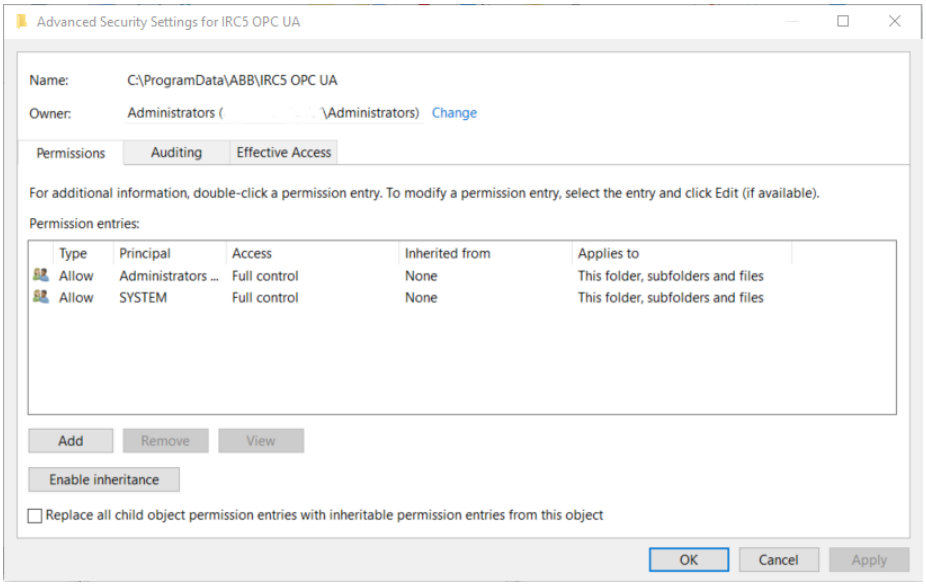
Overview

The IRC5 OPC UA Server keeps several files in a folder called “IRC5 OPC UA” under %ProgramData%\ABB[1]. These files include configuration, log and certificate files, and should be protected from unintentional or malicious access. During initial installation, this folder is created, and security settings are configured to allow access by users belonging to the Administrators group and the SYSTEM user only. It is possible to change these security settings manually to provide access to other users. Any changes to the security settings are preserved during an upgrade of the IRC5 OPC UA Server.

Procedure to view security settings

To view the security settings, do the following:

1. Navigate to the **%ProgramData%\ABB** folder in Windows Explorer.
2. Right-click on the **IRC5 OPC UA** folder and select **Properties.**
3. In the **Properties** dialog, select the **Security** tab, and press the **Advanced** button.
4. The **Advanced Security Settings** dialog may show a **Continue** button to attempt the operation with administrative permissions. Click the **Continue** button and enter administrative credentials if asked for. If successful, the dialog shown in the screen shot below should appear.



xx2000000901

To restore the security settings to the recommended, default values, do the following:

1. Make sure inheritance is turned off (press the **Disable inheritance** button, if shown).

1 %ProgramData% is by default C:\ProgramData, but may be different depending on the Windows installation.

Continues on next page

3.3.3 Restricting Access to application folder
Continued

2 Modify the list of permission entries to include SYSTEM and Administrators only.
3 Tick the check box for **Replace all child object permissions entries from this object**.
4 Click **Apply** button.

3.4 How to Connect to IRC5 OPC UA Server

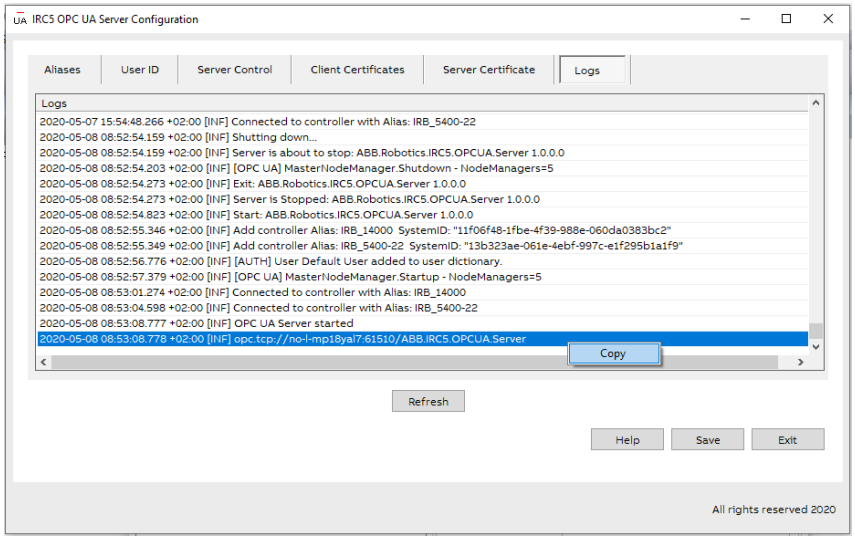
Overview

In a secure connection, the OPC UA server and OPC UA client must trust each other to protect the data exchange between the OPC UA server and OPC UA client. To establish a secure connection between an OPC UA server and OPC UA client, complete the following tasks:

1 Endpoint Information

- Endpoint URL:

  The easiest way to get the correct endpoint URL is to go to the **Log** tab in the IRC5 OPC UA Config Tool, find the line containing the server's endpoint URL and right-click it to copy. The server's endpoint URL will not change unless the name of the computer the server is running on is changed, or the port number is changed using the IRC5 OPC UA Server Config Tool.



xx2000001096

- Security Settings

  Security Policy: Select one of the following IRC5 OPC UA Server supported security policies.

  - Basic256Sha256
  - Aes128_Sha256_RsaOaep
  - Aes256_Sha256_RsaPss

  Message Security Mode (or) Security Policy: Select one of the following IRC5 OPC UA Server supported security mode

  - Sign
  - Sign and encrypt

  Security Message Encoding: Select **Binary** as it is supported by IRC5 OPC UA Server.

2 Authentication Settings

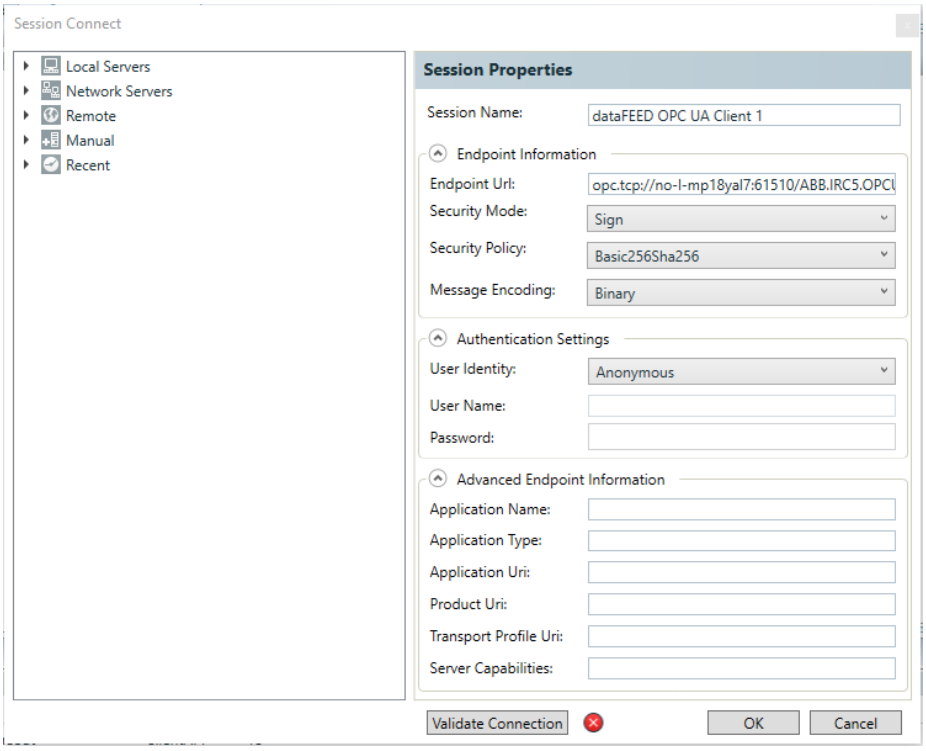
Continues on next page

Select one of the IRC5 OPC UA Server supported user token type (User Identity)

- Anonymous: No user information is available.
- User Name: A user is identified by username and password.

  OPC UA Clients should provide RobotWare username and password for authentication.

Ex: We are using Softing's dataFEED OPC UA test client as an example



xx2000001099

3 The first time a connection attempt made from a new OPC UA client, it will fail.

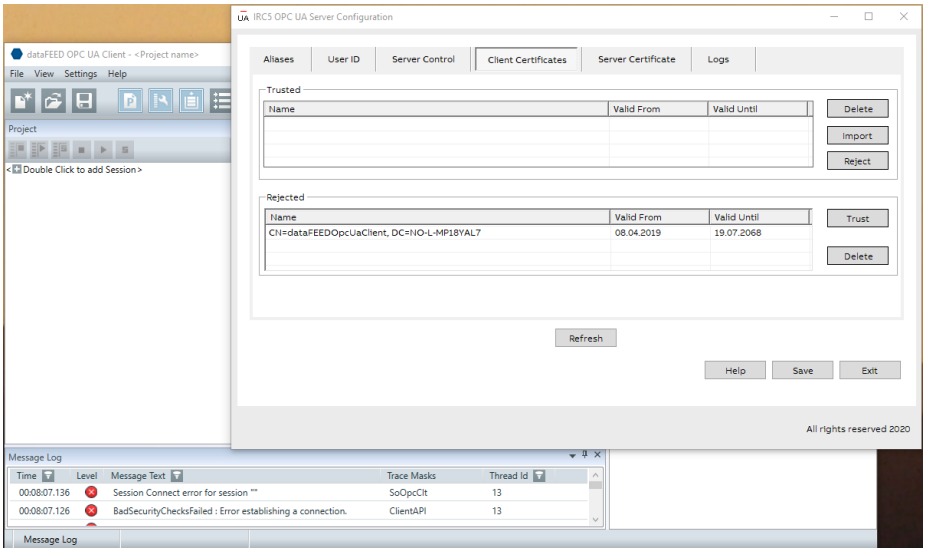
  This is for security reasons; OPC UA clients and servers may use certificates to make sure they communicate with an approved server or client, and the default security settings for the IRC5 OPC UA Server is to require a trusted client certificate. In addition, firewall settings may also cause the connection to fail.

4 Trust OPC UA Client Certificate:

  OPC UA Client may gives a **BadSecurityChecksFailed** error message. Go to the **Client Certificates** tab in the IRC5 OPC UA Config Tool and check for new certificate in the **Rejected** list. Select this certificate and press the **Trust** button. This moves the certificate to the **Trusted** list.

Continues on next page

When a client tries to connect for the first time, it will be rejected and it's certificate will be placed in the **Rejected** list. This allows the administrator to review clients before allowing them to connect.



xx2000001097

If this does not help, check for firewall log and firewall settings and consider adding a rule for allowing incoming connections to the port number specified in the endpoint URL.

5 Trust IRC5 OPC UA Server Certificate: OPC UA Clients should also trust the IRC5 OPC UA Server certificate to establish secure communication.

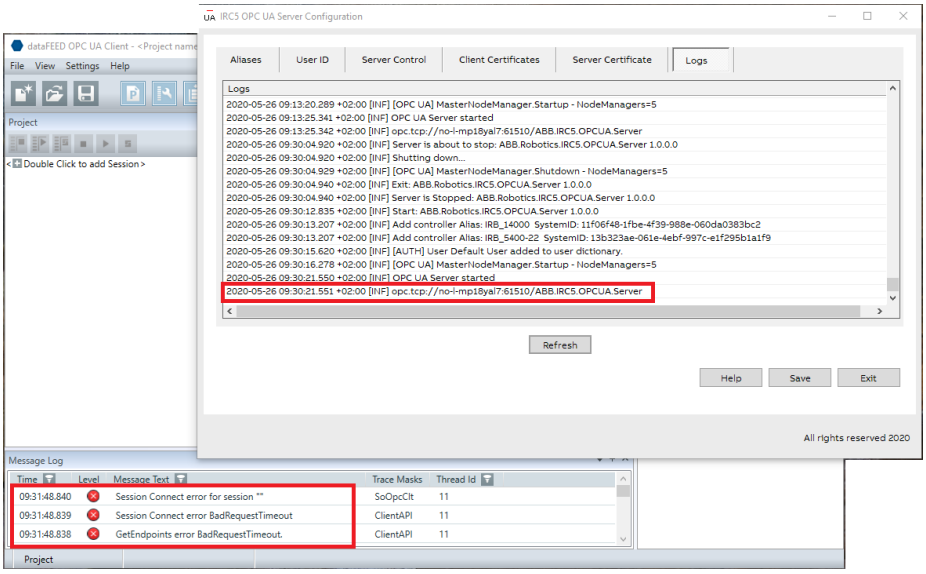
6 Firewall issues :

If the OPC UA client complains about timeout connection errors, and there is nothing in the IRC5 OPC UA Server log about failed connection attempts, it is most likely that the firewall settings needs to be changed.

How to change the firewall settings depends on the firewall solution. Consult IS-support to create a firewall rule that allows incoming connections to the

Continues on next page

IRC5 OPC UA Server. The port number used by the IRC5 OPC UA Server can be seen in the **Server Control** tab of the configuration tool.



xx2000001098

Note

First, check that the IRC5 OPC UA Server is running. This is easy to do by opening the Windows **Task Manager** and select the **Services** tab. The *ABB.Robotics.IRC5.OPCUA.Server* should have a **Running** status. If it is stopped, right click and select **Start** to start it again.

4 Troubleshooting

Overview

The following table provides some helpful information for troubleshooting:

Behavior	Notes
An OPC UA Client cannot connect to the IRC5 OPC UA server although the server is running as windows service. The error message **Host unreachable** (or similar) appears.	Check whether firewall settings prevent communication with the IRC5 OPC UA server. The server port must be open for incoming TCP communication so that a client can connect. The server port is configurable and the current setting can be found in the Server Control tab of the IRC5 OPC UA Server Config Tool.
An OPC UA Client sees the server's endpoints, but a connection with them fails with the error message **Host unreachable**.	1 Check that the name resolution in your network is working properly and that the server is accessible under its host name. Even if the OPC UA Client apparently connects to the IP address of the server (e.g. *opc.tcp://192.168.0.1:65150*) to access the OPC UA server's endpoints, the server always returns its own host name in its endpoints. If the client connects directly to one of the endpoints, it will use the host name of the server again. If the name resolution does not work, the connection fails. 2 Try using hostname in endpoint URL Example: *opc.tcp://MyOPCUAServer-Host:61510/ABB.IRC5.OPCUA.Server*
An OPC UA Client sees the endpoints of the server, but a connection to a secure endpoint fails. The error message **BadSecurityChecksFailed** appears.	Check whether the server trusts the client certificate. The required configuration steps can be found in section Trust the rejected client certificate(s).
An OPC UA Client sees the endpoints of the server, but a connection to a secure endpoint fails. The error message **BadSecurityPolicyRejected** appears.	IRC5 OPC UA Server rejects the client connection because of unsupported security policy provided by client. OPC UA Client should select the security polices that is supported by IRC5 OPC UA Server. For more information see *Security configuration on page 53* or *How to Connect to IRC5 OPC UA Server on page 57*.

This page is intentionally left blank

5 Appendix

5.1 Appendix A - ABB Robotics OPC UA proprietary information model for IRC5

5.1.1 Overview

This section describes the OPC UA information model for ABB robot controllers.

5.1.2 OPC Unified Architecture for ABB Robotics Controller

ObjectType RobotControllersType

A container for Robot Controller objects.

Table 1: ObjectType RobotControllersType

Attribute	Value
BrowseName	RobotControllersType
IsAbstract	False

Subtype of FolderType of http://opcfoundation.org/UA/

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasComponent	Object	S_Alias_name_		RobotControllerType	OptionalPlaceholder	

S_Alias_name_: A robot controller is identified by its alias name that must be unique.

ObjectType RobotControllerType

Top level object type for an ABB Robotics Controller.

Table 2: ObjectType RobotControllerType

Attribute	Value
BrowseName	RobotControllersType
IsAbstract	False

Subtype of BaseObjectType of http://opcfoundation.org/UA/

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasProperty	Property	BootVersion	String	PropertyType	Mandatory	Read
HasProperty	Property	ControllerAddress	String	PropertyType	Mandatory	Read
HasProperty	Property	ControllerID	String	PropertyType	Mandatory	Read
HasProperty	Property	ControllerName	String	PropertyType	Mandatory	Read
HasProperty	Property	SystemID	Guid	PropertyType	Mandatory	Read
HasProperty	Property	SystemName	String	PropertyType	Mandatory	Read
HasComponent	Variable	CollisionDetectState	CollisionDetectStateEnum	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	ControllerExecutionState	ControllerExecutionStateEnum	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	ControllerState	ControllerExecutionStateEnum	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	InterfaceState	InterfaceStateEnum	BaseDataVariableType	Mandatory	Read

Continues on next page

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasComponent	Variable	OperatingMode	OperatingModeEnum	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	SpeedRatio	Int32	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	SystemClock	DateTime	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	RapidProgramUsedMemory	UInt32	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	RapidProgramFreeMemory	UInt32	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	MasterRAPID	MastershipEnum	BaseDataVariableType	Mandatory	Read
HasComponent	Variable	MasterCFG	MastershipEnum	BaseDataVariableType	Mandatory	Read
HasComponent	Object	IO_System		IOSystemType	Mandatory	
HasComponent	Object	RAPID		RAPIDType	Mandatory	

BootVersion: A read-only string that contains the value of the robot controller's RobotWare operating system version.

ControllerAddress: A read-only string that specifies either a) the IP network address of the Real Controller (RC), or b) the directory path to the RobotWare system active in the Virtual Controller (VC) running on the PC.

ControllerID: By default, set to the serial number of the controller and is thereby a unique identifier of the controller. This is a read-only value.

ControllerName: An identification of the controller that is independent of the system or the software running on the controller. This name comes from the robot controller and may be the same as the AliasName, however while the AliasName must be unique, there is no such requirement on the ControllerName. This is a read-only value.

SystemID: A read-only GUID that contains the identifier that globally and uniquely identifies a robot controller/system combination.

SystemName: A read-only string that contains the name of the RobotWare system currently loaded. This is the name assigned by the user when creating a system in e.g. Installation Manager.

CollisionDetectState: A read-only value that contains the state of the collision detection mechanism in the robot controller. See the definition of CollissionDetectStateEnum for details.

ControllerExecutionState: A read-only value that contains the execution state (Running or Stopped) of the robot controller.

ControllerState: A read-only value that contains the state of the robot controller. See the definition of ControllerStateEnum for details.

Continues on next page

InterfaceState: A read-only value indicating the state of the communication interface to the robot controller. This state is maintained by the OPC UA server. See the definition of InterfaceStateEnum for details.

OperatingMode: A read-only value that contains the robot controller operational mode. See the definition of OperationgModeEnum for details.

SpeedRatio: A read-only value that defines the speed ratio of the robot controller in percent, range 0 - 100.

SystemClock: A read-only value that contains the robot controller's system clock value. It is only valid when the interface to the controller is operational.

RapidProgramUsedMemory: A read-only value that defines the amount of memory in bytes being used by the robot controller's RAPID program.

RapidProgramFreeMemory: A read-only value that defines the amount of memory in bytes available to the robot controller's RAPID program.

MasterRAPID: A read-only value that shows if the mastership of RAPID is held by another client. See definition of MastershipEnun for details.

MasterCFG: A read-only value that shows if the mastership of CFG is held by another client. See definition of MastershipEnun for details.

IO_System: Represents the I/O system of the controller.

RAPID: Container for all RAPID tasks in the controller.

ObjectType IOSystemType

Object type describing the the I/O system of the robot controller.

Table 3: ObjectType IOSystemType

Attribute	Value
BrowseName	IOSystemType
IsAbstract	False

Subtype of FolderType of http://opcfoundation.org/UA/

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasComponent	Object	IO_Signals		IOSignalsType	Mandatory	

IO_Signals: Container for all I/O signals in the controller.

ObjectType IOSignalsType

A container for I/O signals.

Table 4: ObjectType IOSignalsType

Attribute	Value
BrowseName	IOSignalsType
IsAbstract	False

Continues on next page

Subtype of FolderType of http://opcfoundation.org/UA/

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasComponent	Variable	S_Signal_name_	BaseDataType	DataItemType	OptionalPlaceholder	ReadWrite

S_Signal_name_: Represents an IO signal. Clients can both read and write output signals. A sucessful write to an output signal requires that the I/O signal's Access Level is set to ALL. Access Level is a system parameter for the I/O signal and can be set either in the I/O configuration file before loading it into the system or using RobotStudio or the FlexPendant.

Note

It is only possible to write to output signals. You must not write to input signals.

For more information on system parameters and the parameter **Access Level**, please refer to*Technical reference manual - System parameters*, **section I/O**.

ObjectType RAPIDType

Object type describing the RAPID sub-system of the robot controller.

Table 5: ObjectType RAPIDType

Attribute	Value
BrowseName	RAPIDType
IsAbstract	False

Subtype of FolderType of http://opcfoundation.org/UA/

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasComponent	Object	S_Task_name_		RAPIDTaskType	MandatoryPlaceholder	

S_Task_name_: Represents a RAPID task in the controller.

ObjectType RAPIDTaskType

Represents a RAPID task in the controller. It acts as a container for any modules loaded in the task.

Table 6: ObjectType RAPIDTaskType

Attribute	Value
BrowseName	RAPIDTaskType
IsAbstract	False

Subtype of FolderType of http://opcfoundation.org/UA/

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasComponent	Variable	TaskExecutionState	TaskExecutionStateEnum	BaseDataVariableType	Mandatory	Read

Continues on next page

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasComponent	Variable	TaskState	TaskExecutionStateEnum	BaseDataVariableType	Mandatory	Read
HasComponent	Object	S_Module_name_		RAPIDModuleType	OptionalPlaceholder	

TaskExecutionState: A read-only value that contains the execution state of the RAPID task. See the definition of **TaskExecutionStateEnum** for details.

TaskState: A read-only value that contains the state of the RAPID task. See definition of TaskStateEnum for details.

S_Module_name_: Represents a RAPID module in a RAPID task.

ObjectType RAPIDModuleType

An object representing a RAPID module. It acts as a container for all persistent variables in the module.

Table 7: ObjectType RAPIDModuleType

Attribute	Value
BrowseName	RAPIDModuleType
IsAbstract	False

Subtype of FolderType of http://opcfoundation.org/UA/

Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule	Access
HasComponent	Variable	S_PERS_name_	BaseDataType	DataItemType	OptionalPlaceholder	ReadWrite

S_PERS_name_: Represents a persistent (PERS) variable in a RAPID module. Clients can both read and write persistent variables. A successful write to a persistent variable requires that no other client has mastership of RAPID. See description of **MasterRAPID** variable.

CollisionDetectStateEnum Values

Defines possible states of the collision detection mechanism in the robot controller.

Table 8: CollisionDetectStateEnum Values

Subtype of Enumeration of http://opcfoundation.org/UA/

Name	Value	Comment
Unknown	0	Unknown.
Initiated	1	Collision detection has been initiated.
Started	2	Collision detection has been started.
Confirmed	3	Collision detected/confirmed.
Acknowledged	4	Collision detected and acknowledged.

Continues on next page

ControllerExecutionStateEnum Values

Defines possible execution states of the robot controller.

Table 9: ControllerExecutionStateEnum Values

Subtype of Enumeration of http://opcfoundation.org/UA/

Name	Value	Comment
Unknown	0	Status is unknown.
Running	1	At least one normal RAPID task is executing or performing regain.
Stopped	2	No normal RAPID task is executing or performing regain.

ControllerStateEnum Values

Defines possible states of the robot controller.

Table 10: ControllerStateEnum Values

Subtype of Enumeration of http://opcfoundation.org/UA/

Name	Value	Comment
Init	0	Initialize state.
MotorsOff	1	Motors off state.
MotorsOn	2	Motors on state.
GuardStop	3	Guard stop state.
EmergencyStop	4	Emergency stop state.
EmergencyStopReset	5	Emergency stop reset state.
SystemFailure	6	System failure state.
Unknown	99	Unknown state.

InterfaceStateEnum Values

Defines possible states of the interface to the robot controller.

Table 11: InterfaceStateEnum Values

Subtype of Enumeration of http://opcfoundation.org/UA/

Name	Value	Comment
Disconnected	0	The interface to the robot controller is disconnected and non-functional.
Connected	1	The interface to the robot controller is connected and operational.
NoPCInterfaceOption	2	The robot controller does not have the PC Interface RobotWare option that creates the interface to the controller.
UnresolvableAlias	3	The system cannot resolve the indicated alias to a single robot controller on the network.

Continues on next page

OperatingModeEnum Values

Defines possible operational modes of the robot controller.

Table 12: OperatingModeEnum Values

Subtype of Enumeration of http://opcfoundation.org/UA/

Name	Value	Comment
Auto	0	Automatic mode (production).
Init	1	Initialize mode.
ManualReduced-Speed	2	Manual reduced speed mode.
ManualFullSpeed	3	Manual full speed mode.
AutoChange	4	A change to automatic mode has been requested.
ManualFullSpeed-Change	5	A change to manual full speed has been requested.
NotApplicable	6	Controller operating mode is not applicable in current controller state.

TaskExecutionStateEnum Values

Defines possible task execution states.

Table 13: TaskExecutionStateEnum Values

Subtype of Enumeration of http://opcfoundation.org/UA/

Name	Value	Comment
Ready	0	The task has no PCP or execution context.
Stopped	1	Task is not executing or not performing regain. PCP and execution context are defined in task.
Running	2	Task is executing or performing regain.
UnInitiated	3	The program server is not initialized. State only assumed during startup.
Unknown	4	Status is unknown.

TaskStateEnum Values

Defines possible task states.

Table 14: TaskStateEnum Values

Subtype of Enumeration of http://opcfoundation.org/UA/

Name	Value	Comment
Empty	0	No modules are loaded in the task.
Loaded	1	Modules are loaded, but not linked.
Linked	2	Modules are loaded and linked.
Initiated	3	The program server is not initialized. State only assumed during startup.

Continues on next page

MastershipEnum Values

Defines possible mastership values.

Table 15: MastershipEnum Values

Subtype of Enumeration of http://opcfoundation.org/UA/

Name	Value	Comment
NoMaster	0	No client has mastership
HeldRemote	1	A remote client has mastership.
HeldLocal	2	A local client has mastership (typically the TPU)
HeldInternal	3	The controller itself has mastership.

5.2 Appendix B - Performance tests summary

Overview

IRC5 OPC UA Server running on PC with following specifications:

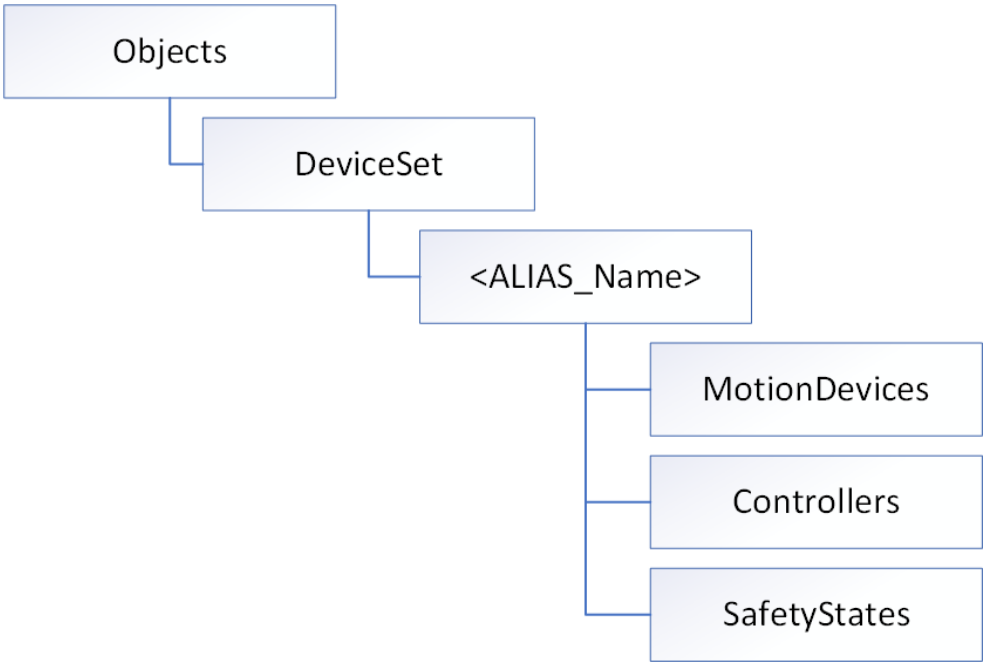
- OS: Windows 10 Pro
- System Type: 64-Bit OS, x64-based processor
- Processor: Intel i5-6500T CPU @ 2.5GHz
- RAM: 8 GB

Sl. No.	Test case description	Test Result
1	IRC5 OPC UA Server with one robot controller having 11 subscription changes per second (7 Rapid variables + 3 output signals + 1 event log)	Pass
2	IRC5 OPC UA Server with one robot controller having 31 subscription changes per second (27 Rapid variables + 3 output signals + 1 event log)	Pass
3	IRC5 OPC UA Server with one robot controller having 51 subscription changes per second (47 Rapid variables + 3 output signals + 1 event log)	Pass
4	IRC5 OPC UA Server with one robot controller having 71 subscription changes per second (67 Rapid variables + 3 output signals + 1 event log)	Pass
5	IRC5 OPC UA Server with one robot controller having 101 subscription changes per second (97 Rapid variables + 3 output signals + 1 event log	Pass
6	IRC5 OPC UA Server with one robot controller having 151 subscription changes per second (147 Rapid variables + 3 output signals + 1 event log)	Pass
7	IRC5 OPC UA Server with one robot controller having 201 subscription changes per second (197 Rapid variables + 3 output signals + 1 event log)	Pass
8	IRC5 OPC UA Server with one virtual controller having 1000 I/O signals and 1000 I/O signals subscription changes per second	Pass
9	IRC5 OPC UA Server with four virtual controllers each having 13 subscription changes per second (10 I/O signals + 3 Rapid variables)	Pass
10	IRC5 OPC UA Server with eight virtual controllers each having 13 subscription changes per second (10 I/O signals + 3 Rapid variables)	Pass
11	IRC5 OPC UA Server with twelve virtual controllers each having 13 subscription changes per second (10 I/O signals + 3 Rapid variables)	Pass

5.3 Appendix C - Robotics companion specification

Introduction

The IRC5 OPC UA Server supports all mandatory and some of the optional parts of **OPC 40010-1 OPC UA for Robotics, Part 1: Vertical Integration**. This chapter describes the supported features and how they map to RobotWare.



xx2000002342

Supported features

DeviceSet

The DeviceSet is a container for all instances of ComponentType defined in OPC Unified Architecture for Devices (DI). One of the subtypes of the ComponentType is the MotionDeviceSystemType as described below.

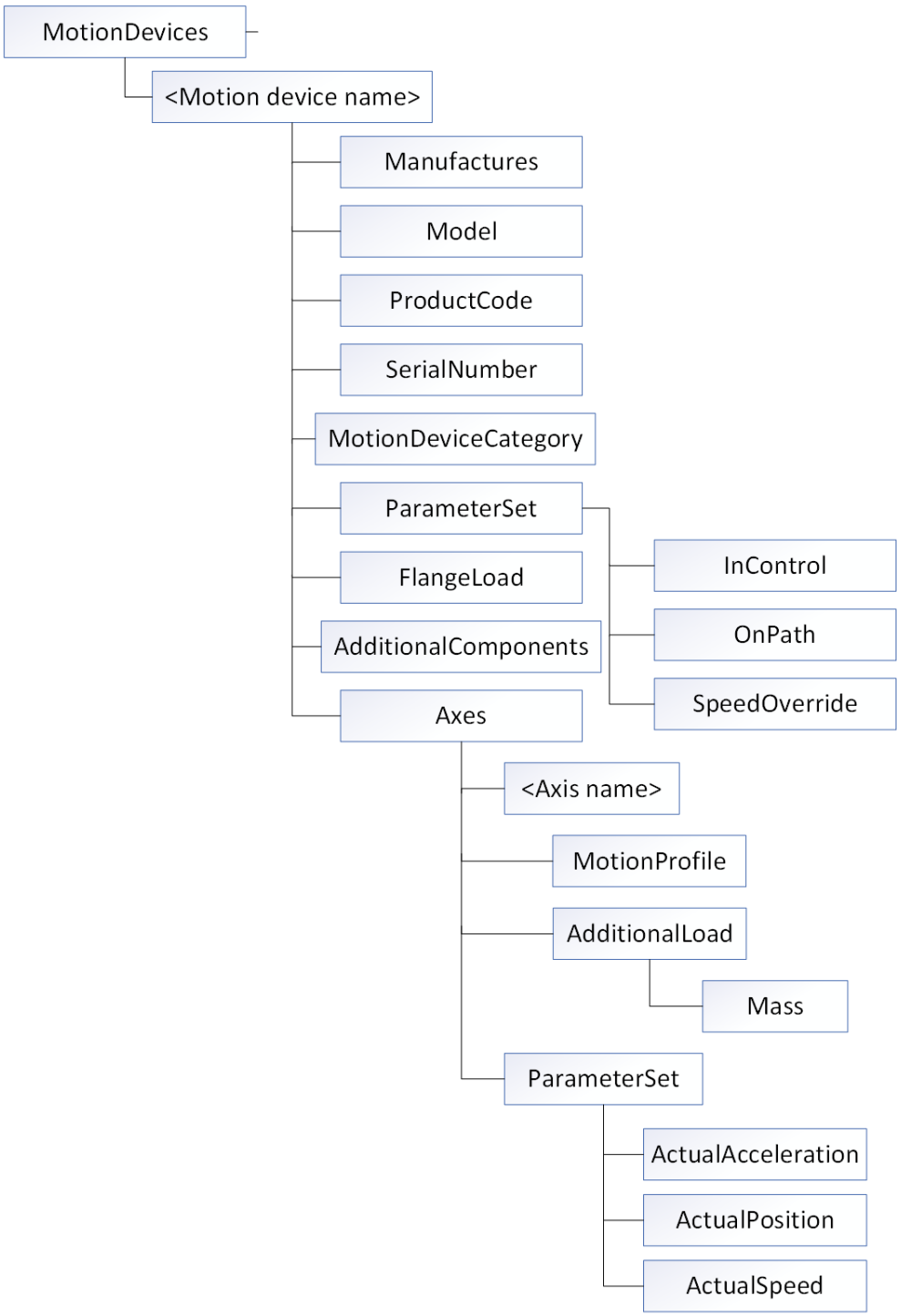
Feature	Browse Name	Description
MotionDeviceSystem	<name>	Each instance corresponds to an ABB robot and <name> equals the Alias name given to the robot in the OPC UA server configuration.

MotionDeviceSystem

Feature	Browse Name	Description
MotionDevices	MotionDevices	A container for instances of MotionDevice-Type
Controllers	Controllers	A container for instances of ControllerType
SafetyStates	SafetyStates	A container for instances of SafetyStateType

Continues on next page

MotionDevices



xx2000002343

Feature	Browse Name	Description
MotionDevice	<name>	Each MotionDevice instance corresponds to a Mechanical Unit in the ABB robot. <name> is equal to the name of the Mechanical Unit.

Continues on next page

MotionDevice

Feature	Browse Name	Description
MotionDeviceCategory	MotionDeviceCategory	A categorization of the type of motion device based on ISO 8373, e.g. ARTICULATED_ROBOT
Manufacturer	Manufacturer	Name of manufacturer, i.e. “ABB”
Model	Model	Maps to the Model property of the Mechanical Unit, e.g. IRB5500_HWT
ProductCode	ProductCode	The article number for the Mechanical Unit, if available. Empty string otherwise.
SerialNumber	SerialNumber	The serial number if the Mechanical Unit, if available. Empty string otherwise.
FlangeLoad	FlangeLoad/Mass	The current Payload Mass of the Mechanical Unit.
ParameterSet	ParameterSet/OnPath	Not supported – always (null)
	ParameterSet/InControl	“true” if Motors ON, “false” otherwise
	ParameterSet/SpeedOverride	The Speed Ratio of the system 0 – 100%
Axes	Axes	A container for instances of AxisType
PowerTrains	PowerTrains	A container for instances of PowerTrainType
AdditionalComponents	AAdditionalComponents	Empty folder, not in use.

Axes

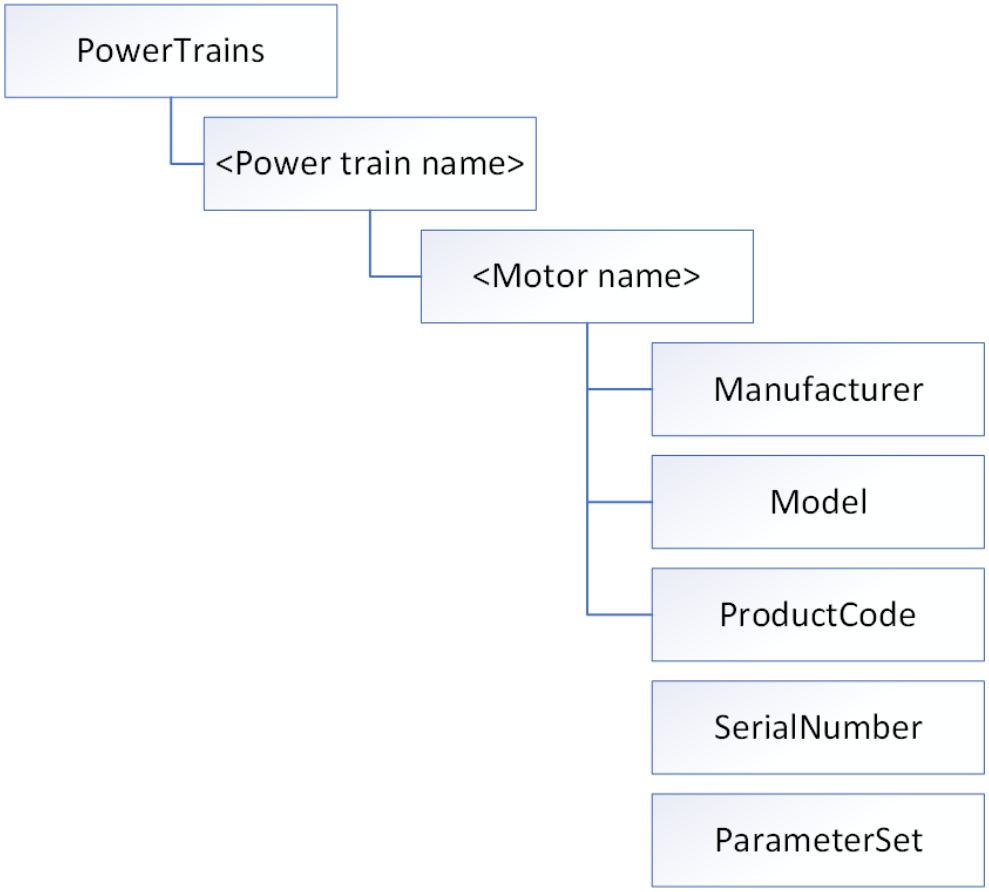
Feature	Browse Name	Description
Axis	<name>	Each instance corresponds to an axis of the Mechanical Unit. <name> is equal to the name of the axis, e.g. Rax_1 or Eax_6

Axis

Feature	Browse Name	Description
MotionProfile	MotionProfile	Property describing the type of motion for this axis, e.g. “ROTARY”.
AdditionalLoad	AdditionalLoad/Mass	Not supported – always 0.0
ParameterSet	ParameterSet/ActualPosition	Current position of axis
	ParameterSet/ActualSpeed	Not supported – always (null)
	ParameterSet/ActualAcceleration	Not supported – always (null)

Continues on next page

PowerTrains



xx2000002344

Feature	Browse Name	Description
PowerTrain	<name>	Each instance corresponds to a PowerTrain of the Mechanical Unit. <name> is equal to the joint name of the robot or the external axis of the mechanical unit that the power train drives. E.g. rob_1_1

PowerTrain

Feature	Browse Name	Description
Motor	<name>	Each instance of the MotorType corresponds to a Motor of the Axis. Normally there is one Motor per Axis. <name> is equal to the name of the Power-Train instance it belongs to.

Motor

Feature	Browse Name	Description
Manufacturer	Manufacturer	Name of manufacturer, i.e. “ABB”
Model	Model	Not supported – always (null)

Continues on next page

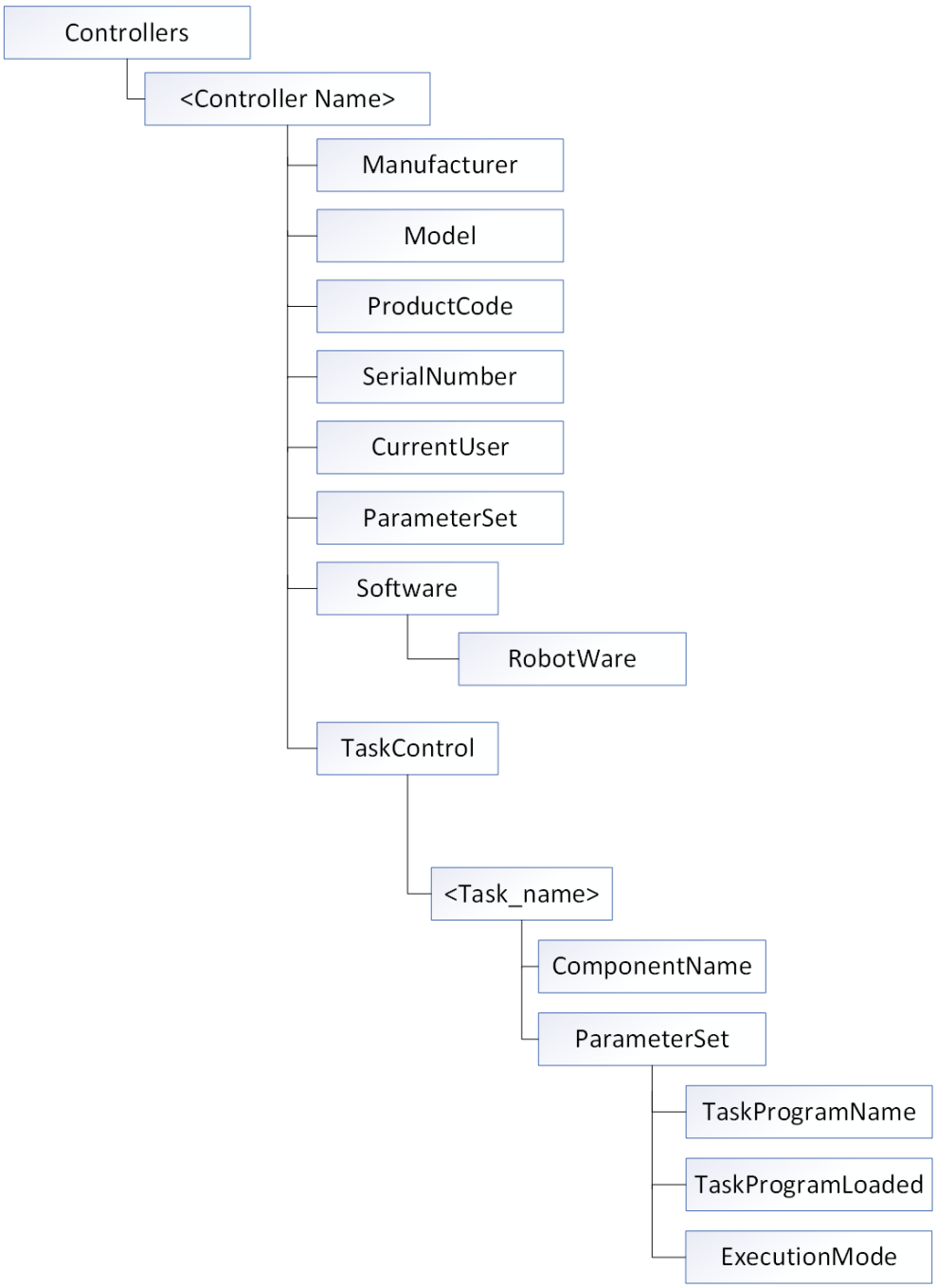
Feature	Browse Name	Description
ProductCode	ProductCode	Article number of Motor, retrieved from the “Use Motor Type” field of the Motor configuration.
SerialNumber		Not supported – always (null)
ParameterSet	ParameterSet/BrakeReleased	Not supported – always (null)
	ParameterSet/MotorTemperature	Not supported – always (null) This is a mandatory variable, but as ABB robots have only PTCs and not analog temperature sensors in the motors, there is no temperature to read.
	ParameterSet/EffectiveLoadRate	Not supported – always (null)

Continues on next page

5.3 Appendix C - Robotics companion specification

Continued

Controllers



xx2000002345

Feature	Browse Name	Description
Controller	<name>	There is always only one instance of ControllerType for ABB systems. <name> is equal to the Controller Name from the Controller Properties

Continues on next page

Controller

Feature	Browse Name	Description
Manufacturer	Manufacturer	Name of manufacturer, i.e. “ABB”
Model	Model	Name of controller model, e.g. “IRC5”
ProductCode	ProductCode	Article number of controller. Not available digitally, so value is set to empty string.
SerialNumber	SerialNumber	Serial number of controller. Not available digitally, so value is set to empty string.
CurrentUser	CurrentUser/Level	String containing list of grants assigned to the current user.
	CurrentUser/Name	Name of current user, e.g. “Default User”
ParameterSet	ParameterSet/TotalPowerOnTime	Not supported – always (null)
	ParameterSet/StartUpTime	Not supported – always (null)
	ParameterSet/UpsState	Not supported – always (null)
	ParameterSet/TotalEnergyConsumption	Not supported – always (null)
	ParameterSet/CabinetFanSpeed	Not supported – always (null)
	ParameterSet/CPUFanSpeed	Not supported – always (null)
	ParameterSet/InputVoltage	Not supported – always (null)
	ParameterSet/Temperature	Not supported – always (null)
Components	Components	Empty folder, not in use.
Software	Software	A container for instances of SoftwareType
TaskControls	TaskControls	A container for instances of TaskControlType

Software

Feature	Browse Name	Description
Software	<name>	A list of software on the robot controller. For ABB controllers this list contains only one instance named RobotWare

Software: RobotWare

Feature	Browse Name	Description
Manufacturer	Manufacturer	Name of manufacturer, i.e. “ABB”
Model	Model	Name of software, typically “RobotWare”
SoftwareRevision	SoftwareRevision	Version number of software, e.g. “6.11.0.1”

Continues on next page

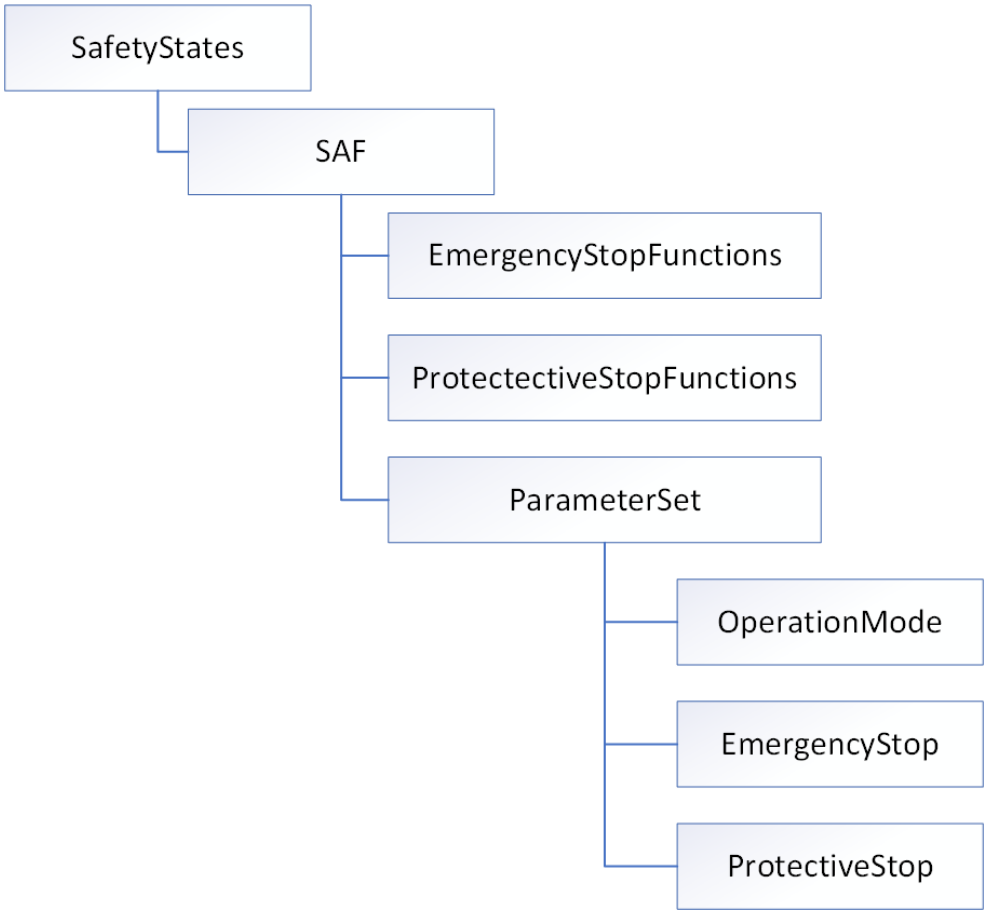
TaskControls

Feature	Browse Name	Description
TaskControl	<name>	Each TaskControlType instance listed corresponds to a RAPID task. <name> is equal to the name of the corresponding RAPID task.

TaskControl

Feature	Browse Name	Description
ComponentName	ComponentName	Contains the name of the corresponding RAPID task, i.e. the same as the Browse Name for the TaskControl itself.
ParameterSet	ParameterSet/TaskProgramName	
	ParameterSet/TaskProgramLoaded	True if the RAPID task has a defined execution context, i.e. that some RAPID code is loaded and the program pointer is defined.
	ParameterSet/ExecutionMode	Not supported – always (null)

SafetyStates



xx2000002346

Continues on next page

Feature	Browse Name	Description
SafetyState	<name>	For ABB controllers this list contains only one instance named “SAF”

SafetyState: SAF

Feature	Browse Name	Description
ParameterSet	ParameterSet/OperationalMode	Corresponds to the Operating Mode of the robot controller.
	ParameterSet/EmergencyStop	True if emergency stop is activated, false otherwise.
	ParameterSet/ProtectiveStop	True of one or more protective stops are activated, false otherwise.
EmergencyStopFunctions	EmergencyStopFunctions	Empty folder, not in use.
ProtectiveStopFunctions	ProtectiveStopFunctions	Empty folder, not in use.

This page is intentionally left blank

Index

ABB AB
Robotics & Discrete Automation
S-721 68 VÄSTERÅS, Sweden
Telephone +46 (0) 21 344 400

ABB AS
Robotics & Discrete Automation
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.
Robotics & Discrete Automation
No. 4528 Kangxin Highway
PuDong District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

ABB Inc.
Robotics & Discrete Automation
1250 Brown Road
Auburn Hills, MI 48326
USA
Telephone: +1 248 391 9000

abb.com/robotics

3HAC074394-001, Rev B, en