UNS Data Servers Reference

Complete configuration options for Data Server services.

Technical Reference UNS Reference UNS Data Servers Reference

Overview

OPC UA Server Configuration
MQTT Broker Configuration
MCP for Runtime Configuration
DataHub Service Configuration
Startup Options
Port Summary
Partner APIs
See Also

FrameworX includes built-in services that expose your Unified Namespace (UNS) to external systems. These services run alongside your solution and provide standardized access to live data without additional infrastructure.

Service	Protocol	Purpose
OPC UA Server	OPC UA	Expose UNS to enterprise systems and other SCADA platforms
MQTT Broker	MQTT	Publish/subscribe access for IoT devices and cloud platforms
MCP for Runtime	MCP	Enable AI/LLM models to query live solution data
DataHub Service	TCP	Synchronize data between multiple FrameworX solutions

OPC UA Server Configuration

Location: Unified Namespace Data Servers OPC UA Server [??]

Setting	Description	Default
Port	TCP port for OPC UA connections	4840
Security Mode	None, Sign, SignAndEncrypt	None
User Authentication	Anonymous, Username/Password, Certificate	Anonymous

See OPC UA Server Reference for complete protocol details

MQTT Broker Configuration

Location: Unified Namespace Data Servers MQTT Broker [??]

Setting	Description	Default
TCP Port	Standard MQTT port	1883
WebSocket Port	WebSocket transport port (0 = disabled)	8083
Authentication	Anonymous, Username/Password	Anonymous
TLS	Enable encrypted connections	Disabled

See MQTT Broker Reference for complete protocol details

MCP for Runtime Configuration

Location: Scripts Classes (custom tools)

Setting	Description	Default
Host	Listening address	localhost
Port	MCP server port	5000
Transport	Communication protocol	stdio

Built-in Tools

Tool	Description
Get Value	Read current tag values from UNS
Get Historian	Query historical data
Get Alarms	Retrieve alarm status and history

Custom Tools

Custom tools are defined in Script Classes using the `[McpServerTool]` attribute. The companion `[Description]` attribute supplies the text the AI model reads to decide when to call the tool — on the method, and on each parameter:

```
[McpServerTool, Description("Returns power output for specified solar panel")]
public double runtime_get_panel_power(
[Description("Identifier of the solar panel to query.")] string panelId)
{
    return @Tag.SolarPanels[panelId].Power.Value;
}
```

The method name becomes the tool name published to the AI client, normalized to snake_case. Name methods in snake_case with a `runtime_` prefix so they stay distinguishable from Designer tools in a session that has both connected.

> ⚠ **ScriptsClasses row setup — `ClassContent` must be `MCPTool`.** When a class uses Microsoft MCP SDK attributes (`[McpServerTool]` / `[McpServerToolType]`), set the ScriptsClasses row's `ClassContent` field to `MCPTool`, not `Namespace`. The `MCPTool` branch of the compiler auto-injects the net10 MCP-attribute reference DLLs that those attributes resolve against. The `Namespace` branch ships only the netstandard 2.0 reference set and cannot host the MCP attribute types — using it for an MCP-tool class produces CS0012, CS0246, and CS0653 cascades at build time. Standard **Class with Methods** does not register the row as a tool surface at all and is not a substitute.

See AI MCP for Runtime Connector for setup details See AI MCP for Runtime Tutorial for step-by-step guide

---

DataHub Service Configuration

Location: Unified Namespace Data Servers DataHub [??]

Setting	Description	Default
Listening Port	TCP port for DataHub connections	30001
Access Key	Authentication key for connections	(none)
Access Mode	Read, Write, or ReadWrite	ReadWrite

See Data Hub Communication Service for architecture details

Startup Options

Services are enabled on the **Runtime Startup** page:

Service	Startup Checkbox
OPC UA Server	?? OPCServer
MQTT Broker	?? MQTTBroker
MCP for Runtime	(Enabled via Scripts)
DataHub Service	?? Publish to DataHub

See Data Servers How-to for step-by-step enable procedures

Port Summary

Service	Default Port	Protocol
OPC UA Server	4840	TCP
MQTT Broker (TCP)	1883	TCP
MQTT Broker (WebSocket)	8083	TCP/HTTP
MCP for Runtime	5000	stdio
DataHub Service	30001	TCP

Partner APIs

For development partners requiring direct programmatic access:

API	Description
.NET API	Native .NET library for direct solution integration
REST API	HTTP-based access for web and cloud applications

Contact Tatsoft for partner API access and documentation.

See Also

- Data Servers — Concept overview
- Data Servers How-to — Enable and configure services
- TagProvider Services — Connect to external systems (FrameworX as client)