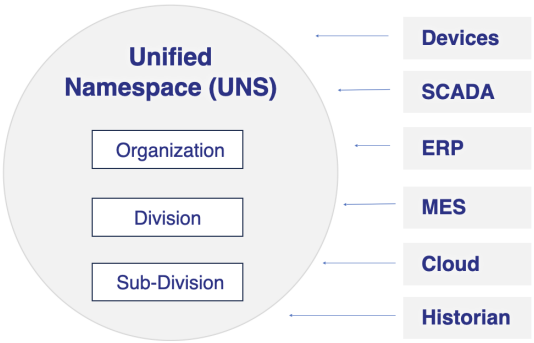
Unified Namespace

Centralized real-time data repository with hierarchical organization for all solution data.

Modules Unified Namespace | Tutorial | How-to Guide | Reference

Unified Namespace Overview

The UNS is the first pillar on the Four Pillars methodology, it provides a centralized real-time data repository with hierarchical organization for all solution data. As the first pillar in the Four-Pillar methodology, it creates a standardized method for accessing and managing tags, assets, and external data sources.



Unified Namespace Overview
Pillar 1: Unified Namespace (Foundation)
Key Concepts
What the UNS Does
Configuration Workflow
Runtime Behavior
Features Highlights
Common Implementation Patterns
Development Efficiency Tools
Unified Namespace Links

Pillar 1: Unified Namespace (Foundation)

Purpose

The Unified Namespace (UNS) is your solution's data foundation - a single source of truth for all real-time and configuration data.

What to Build

Tag Structure	Asset Tree	UserTypes
• Define naming conventions • Create tag hierarchy • Set data types and ranges • Configure engineering units	• Mirror physical/logical structure - • Organize by area/process/equipment - • Create navigable hierarchy - • Enable asset-based navigation	• Create equipment templates • Define standard objects • Build reusable components • Ensure consistency

Key Concepts

Concept	Description	Example
Tag	Process variable representing real-time and historical data	Tank01.Level, Pump01.Status
Asset Tree	Hierarchical representation of solution data	/Plant/Area1/Line1/Equipment
UserType (UDT)	Reusable data structure with variables and references	Motor template with Speed, Current, Status
TagProvider Services	Direct interaction with variables in remote systems	OPC UA server, MQTT broker connections
Enumeration	Named integer values with messages and colors	0=Stopped, 1=Running, 2=Fault
Runtime Database	In-memory real-time database managing UNS data	TServer.exe process

TagProvider Services

Auto-discover and connect to external data sources. Independent from the Devices module, just share round 15-20 protocols which allow discovery services.

Supports two service types:

- Tag Discovery (external data into UNS) and
- Module Extension (third-party modules).

DataLink

Property on a tag that binds it to a path in a TagProvider's discovered namespace. The runtime auto-manages communication on demand. Tags with DataLink set are informally called "Linked Tags."

example:	Tag `Plant/Tank1/Level` with DataLink `MyMQTT.("sensors/tank1/level")`

What the UNS Does

The Unified Namespace serves as your solution's data backbone:

Data Organization	System Integration
• Creates real-time data models with tags and templates • Organizes data in hierarchical Asset Tree structure • Maintains relationships between equipment and data • Provides intuitive navigation through process hierarchy	• Enables dynamic connectivity to external systems • Supports advanced data types including SQL datasets and JSON objects • Provides type-safe references and up to tri-dimensional arrays • Maintains event-driven, in-memory database

---

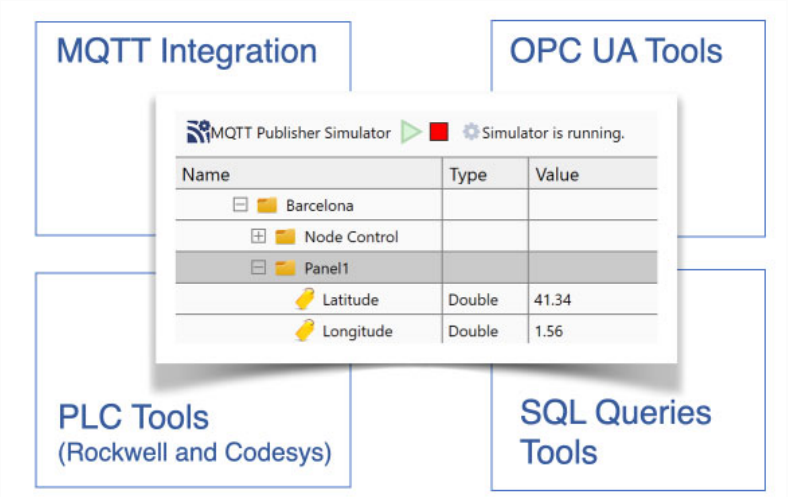
Configuration Workflow

Step	Action	Tool (Designer UI)	Tools
1. Define UserTypes	Create reusable structures	UNS UserTypes	Template Editor
2. Build Asset Tree	Organize hierarchy	UNS Asset Tree	Asset Browser
3. Create Tags	Define variables	UNS Tags	Tag Editor, DataExplorer
4. Configure Enumerations	Set discrete values	UNS Enumerations	Enumeration Editor
5. Add TagProvider Services	Connect external data	UNS TagProvider Services	TagProvider Services
6. Map Remote Data	Link to Asset folders	UNS Asset Tree	Asset Mapping

DataExplorer: Accelerating UNS Development

The DataExplorer tool streamlines UNS creation by enabling:

- **Discovery** - Browse MQTT brokers, OPC servers, PLCs, and SQL databases
- **Import** - Create tags and templates directly from discovered structures
- **Validate** - Test connections before adding to production



This tool reduces manual configuration time by 50-70% when building your tag structure during the Foundation phase.

Learn more in the DataExplorer How-to Guide

Runtime Behavior

Data Loading and Management

When solution execution starts, the Unified Namespace definition loads into the in-memory real-time database (TServer.exe process). This database enables event-driven data exchange between all solution modules with millisecond precision.

Data Access Methods

Modules access Unified Namespace variables through two primary methods:

Direct Tag Access (Local Tags)	Asset Method Access: Using the Asset() function

Using tag names
• Example: `Tag.Machine1.Temperature`
• Supports both local tags and nested UserType properties
 	
When connecting the tag with the field device
• Use the **Devices** Module to map the Tag to a DevicePoint
• If the protocol has TagDiscovery, you can use the **DataLink** field
  ◦ In this case only the TagProvider services must be enabled
	• Example: `Asset("/Line1/Machine1.Temperature")`
	• Local Tags can be also use this syntax
	
	TagProvider TagDiscovery Service
	• If the protocol supports TagDiscovery (OPC UA, MQTT, ..)
	  ◦ Dynamic Tags can created in runtime as a Asset("path") is used.
	  ◦ see [UNS Tag Discovery Reference]
	  ▪ for details on Local Tags, Linked Tags, and Dynamic Tags patterns

If the protocol supports Tag Discovery, you can set the *DataLink* property on the tag to a TagProvider path. The runtime auto-manages communication — no Device Points needed.

See [UNS Tag Discovery Reference] for the full explanation of Linked Tags.

Features Highlights

Advanced Data Types

- **SQL-compatible types** - Full database data type support
- **.NET entities** - Images, JSON, DataTables in single tags
- **Dynamic structures** - Tri-dimensional arrays, lists, collections
- **Type-safe references** - Dynamic tag assignments with validation

Hierarchical Organization

- **Asset folders** - Logical grouping matching equipment
- **Nested templates** - Composition and inheritance support
- **Reusable components** - Template-based design patterns
- **Clear data paths** - Intuitive navigation through hierarchy

External System Integration

- **Protocol support** - ControlLogix, OPC-UA, MQTT SparkPlugB
- **Database connectivity** - SQL, CanaryLabs, InfluxDB, AVEVA PI System
- **Remote applications** - Other FrameworX solutions
- **Dynamic browsing** - Real-time discovery of data structures

Real-time Performance

- **Event-driven updates** - Millisecond timestamp precision
- **In-memory database** - High-performance data access
- **Publish-subscribe model** - Efficient module communication
- **Store-and-forward** - Reliable handling during disconnections

---

Common Implementation Patterns

ISA-95 Model Implementation

Level	Asset Structure	Typical Tags
Enterprise	/Company	KPIs, Financial metrics
Site	/Company/Site1	Production totals, Energy usage
Area	/Company/Site1/Area1	Area production, Quality metrics
Line	/Company/Site1/Area1/Line1	Line speed, Product counts
Cell	/Company/Site1/Area1/Line1/Cell1	Equipment status, Process values

Naming Convention Best Practices

Naming Individual Tag Names

```
Standard Format: [Area]_[Equipment]_[Component]_[Signal]

Examples:
WTP_PUMP01_MOTOR_RUNNING
WTP_PUMP01_MOTOR_SPEED_SP
WTP_TANK01_LEVEL_PV
BLDG_HVAC_AHU01_TEMP_SP

if using UDT SP is BLDG_HVAC_AHU01_TEMP.SP
```

Naming using Asset Folders and UDTs

```
Standard Format: [Area]/[Equipment]/[Component][Signal].[Attribute]

Examples:
WTP/PUMP01/MOTOR/RUNNING
WTP/PUMP01/MOTOR/SPEED.SP
WTP/TANK01/LEVEL.PV
BLDG/HVAC/AHU01/TEMP.SP
```

Naming adding a DataTemplate for Components(MOTOR, TANK)

```
Standard Format: [Area]/[Equipment]/[ComponentUDT].[Signal]

Examples:
WTP/PUMP01/MOTOR.RUNNING
WTP/PUMP01/MOTOR.SPEED.SP
WTP/TANK01/LEVEL.PV
BLDG/HVAC/AHU01.TEMP.SP
```

Development Efficiency Tools

- **Bulk operations** - Import/export via Excel, CSV
- **Search and filter** - Quick tag location and editing
- **Template library** - Reusable component repository
- **Cross-reference** - Find tag usage across modules
- **Version tracking** - Built-in change management



- **Rename any Tag , anytime** - Without losing configuration integrity and runtime online updates.
- **Cross-reference Tag usage** - on-the-fly update, information is always instantly available

---

Unified Namespace Links

Explanation - about foundational technologies

Platform Overview / Technology / Real-Time Tag Database

Platform Overview / Technology / Native Unified Namespace

Platform Overview / Technology / TagProvider Services

Explanation - to understand concepts

Modules / Data Foundation / Unified Namespace

Tutorials - to learn by doing

Tutorials / Data Foundation / Unified Namespace Tutorial

How-to Guides - to accomplish specific tasks

How-to Guides / Data Foundation / Unified Namespace How-to Guide

Reference - technical details

In this section...