Devices Nodes Reference

Represent field devices as nodes.

Devices Nodes (Reference) represent physical devices or PLCs in the communication network, with each node containing one or more communication points mapped to device addresses.

Device Nodes provide:

- Device representation in channels
- Primary/backup station configuration
- Point organization
- Import capabilities
- Synchronization tracking
- Communication routing

Each node maps to a specific device (PLC, RTU, controller) accessible through its parent channel.

Creating Nodes
Node Properties
Platform-managed columns
Station Configuration
Importing Nodes
Node Organization
Runtime Behavior
Common Configurations
Synchronization
Best Practices Checklist
Troubleshooting
Diagnostics

Creating Nodes

1. Navigate to **Devices Nodes**
2. Click **Plus** icon to add node
3. Configure:
    - **Channel** - Select parent channel
    - **Name** - Unique node identifier
    - **Description** - Documentation
4. Click **OK**

Node Properties

The columns below are the user-editable properties of a node. Additional columns appear in the grid but are written by the platform itself — see **Platform-managed columns** immediately below for those.

Property	Description	Required
Name	Plain identifier for the node (e.g., `DeltaV_Server`). The parent channel relationship lives in the **Channel** column; do NOT encode the channel in the **Name** with dot-notation.	Yes
Channel	Name of the parent `DevicesChannels` row.	Yes
PrimaryStation	Primary device address. Protocol-dependent format — consult the protocol reference for the exact syntax.	Yes
BackupStation	Optional backup device address. Protocol-dependent format.	No
Description	Free-text documentation for this node.	No
Category	Editable classification label you assign to organize and tag nodes by category — the same classification field available on Device Channels and Device Access Types. Categories assigned to a node help filter and organize nodes, for example in the Device module diagnostics views and in Historian and Dataset data views.	No

---

Platform-managed columns

The columns below appear on the **Devices Nodes** grid but are written by the platform itself. They are NOT for hand-editing, scripted edits, or AI edits — treat them as read-only diagnostic surface. The **Tags Import** tool writes the `Sync*` columns; the platform owns the rest.

Column	Meaning	Written by
SyncDate	*Timestamp of the last node-configuration import or synchronization*	Tags Import tool
SyncStation	*Source details of the last import*	Tags Import tool
SyncSettings	*Configuration of the last import*	Tags Import tool

ID	*Internal row identifier*	Platform
VersionID	*Internal version marker*	Platform
LockState	*Internal edit-lock state*	Platform
LockOwner	*Internal lock owner*	Platform
DateCreated	*Internal creation timestamp*	Platform
DateModified	*Internal modification timestamp*	Platform

Station Configuration

Primary Station

Protocol-specific addressing:

- **Modbus TCP**: IP:Port (192.168.1.10:502)
- **OPC UA**: opc.tcp://server:4840
- **Serial**: COM1;19200;8;N;1
- **ControlLogix**: 192.168.1.20;1;0

Backup Station

To add redundancy to a specific node, enable the **Backup Station** column. If the column is not visible, right-click any column header and select it to enable. The configuration is the same as for **PrimaryStation**, if the primary connection fails, the system will attempt the backup station. If the backup station also fails, the system will automatically switch back to the primary, and so on. The failover timeout can be configured in **Devices / Channels / Timeout**.

The following properties can be accessed from anywhere in the software:

- Toggle this property between `true` and `false` to force a switch between the **PrimaryStation** and **BackupStation**:
  `@Device.Node.<NodeName>.ForceSwitch = true;`
- Toggle this property between `true` and `false` to enable or disable automatic switching when the current node encounters an error:
  `@Device.Node.<NodeName>.DisableAutoSwitch = true;`
- Use this property to verify whether the primary or backup station is currently active:
  `@Device.Node.<NodeName>.IsPrimary = true;`
- This property is true while the node communicates with the **BackupStation**:
  `@Device.Node.<NodeName>.IsBackup`
- From FrameworX 10.1.6, this property has its own identifier. It is reserved for three-station failover and reads false in current configurations:
  `@Device.Node.<NodeName>.IsFailOver`

Changed in FrameworX 10.1.6

In 10.1.5, IsFailOver and IsBackup shared one identifier, so a reference to IsFailOver returned the backup status. From 10.1.6, IsFailOver returns the failover flag, which reads false. Use IsBackup to read the backup status, and review scripts, display bindings and reports reading either property.

You can view additional properties available for each node by typing `@Device.Node.<NodeName>.` and pressing **Ctrl + Space** to open IntelliSense.

Importing Nodes

Automatic Import Methods

Source	Description	Supported Protocols
PLC Programs	Import from PLC files	ControlLogix, Siemens
OPC Browse	Discover OPC servers	OPC UA
Online Discovery	Scan network devices	Modbus, BACnet
CSV Import	Bulk configuration	All protocols

Import Process

1. Select channel and protocol
2. Choose import method
3. Configure connection settings
4. Browse/select items
5. Map to local tags

See Importing PLC Addresses for detailed instructions

Node Organization

Hierarchical Structure

```
Channel
  --- Node (Device)
      --- Point 1
      --- Point 2
```

Naming Conventions

- Use descriptive names
- Include location/function
- Maintain consistency
- Avoid special characters

Runtime Behavior

Connection Management

- Nodes inherit channel settings
- Independent connection per node
- Automatic reconnection
- Failover to backup station

Communication Flow

1. Channel establishes connection
2. Node routes to device address
3. Points read/write through node
4. Backup activates on failure

Common Configurations

Modbus TCP Node

```
Channel: ModbusTCP_Main
Name: PLC_Tank_Farm
PrimaryStation: 192.168.1.100:502
BackupStation: 192.168.1.101:502
Description: Tank farm controller
```

ControlLogix Node

```
Channel: EthernetIP_Channel
Name: CLX_Line1
PrimaryStation: 192.168.1.50;1;0
Description: Line 1 ControlLogix
```

OPC UA Node

```
Channel: OPCUA_Channel
Name: OPC_Server1
PrimaryStation: opc.tcp://server:4840
Description: Main OPC UA Server
```

Synchronization

Import Tracking

The `SyncDate`, `SyncStation`, and `SyncSettings` columns are read-only provenance written by the **Tags Import** tool when nodes are imported or re-synchronized. They are not user-editable — do not type into them and do not target them from scripts or AI tools.

- **SyncDate** - Timestamp of the last import
- **SyncStation** - Source details of the last import
- **SyncSettings** - Configuration of the last import

Refresh Options

- Manual re-import
- Scheduled sync
- Change detection
- Version control

Best Practices Checklist

- **Group Related Devices** - One node per PLC/device
- **Configure Backup Stations** - For critical devices
- **Document Thoroughly** - Use description fields
- **Test Import Results** - Verify addressing
- **Use Consistent Naming** - Establish standards
- **Monitor Sync Status** - Track configuration changes
- **Plan IP Addressing** - Organize network topology

Troubleshooting

Node offline:

- Check channel status
- Verify primary station address
- Test network connectivity
- Review backup station

Import failures:

- Verify device accessibility
- Check authentication
- Review protocol version
- Test with simple import

Communication errors:

- Confirm addressing format
- Check firewall rules
- Verify protocol settings
- Test backup failover

Points not updating:

- Check node status
- Verify point configuration
- Review scan groups
- Monitor diagnostics

Diagnostics

Monitor node health:

```
// Node status
bool isOnline = @Device.Node.NodeName.IsOnline;

// Station active
string activeStation = @Device.Node.NodeName.ActiveStation;

// Communication stats
int successCount = @Device.Node.NodeName.SuccessCount;
int errorCount = @Device.Node.NodeName.ErrorCount;

// Last sync
DateTime syncTime = @Device.Node.NodeName.LastSync;
```

In this section...
