Devices AccessTypes Reference

Control how data is read and written.

Devices AccessTypes (Reference) define read/write rules for Device Points, controlling polling rates, triggers, and communication behavior between the platform and field devices.

AccessTypes provide:

- Polling configuration
- Read/write triggers
- Startup behavior
- Event-based communication
- Display-driven updates
- Block command support

Each Device Point uses one AccessType to determine when and how it communicates.

Creating AccessTypes
AccessType Properties
Predefined access types
Common AccessType Patterns
OnDisplayOrServer Mode
Trigger-Based Operations
Performance Optimization
Block Commands
Best Practices Checklist
Troubleshooting
Examples by Industry

Creating AccessTypes

1. Navigate to **Devices AccessTypes**
2. Click **Plus** icon
3. Configure properties
4. Click **OK**

AccessType Properties

Property	Description	Required
ReadPolling	Enable polling mode Always, Never, OnDisplayOrServer	Yes
ReadPollingRate	Update frequency (ms) 100-60000	Yes
ReadTrigger	Conditional read trigger Tag or expression	No
ReadOnStartup	Read on solution start True/False	No
WriteEvents	Enable write capability True/False	Yes
WriteEvent	Write trigger condition OnChange, OnTrigger	No
WriteTrigger	Conditional write trigger Tag or expression	No
AcceptUnsolicited	Accept unrequested data True/False	No
IsDeviceControl	Flags the access type as driving MQTT SparkplugB device-control semantics True/False	No
BlockCommand	Protocol-specific blocks True/False	No
Category	Editable classification label you assign to organize and tag access types by category — the same classification field available on Device Channels and Device Nodes. Categories assigned to an access type help filter and organize access types, for example in the Device module diagnostics views and in Historian and Dataset data views.	No

Predefined access types

Every new solution ships with a seed catalog of six predefined AccessType rows. These cover the most common read and write patterns and are ready to assign to Device Points without any further configuration.

Name	Description
Read	Read using the PollingRate
Write	Write when the tag changes its value
ReadWrite	Read and Write the tag value
Setpoints	Read on Startup, then WriteOnly
DeviceControl	MQTT SparkplugB Publisher Device Control
WriteUpEvent	Write when value changes up, typically for PLC commands using TimerDelayOff tags

You can edit or extend the seed catalog like any other AccessType row. Custom rows you add appear alongside the predefined ones in the AccessType picker on Device Points.

IsDeviceControl vs DeviceControl

The AccessTypes table has both a boolean column named **IsDeviceControl** and a seed row whose Name is **DeviceControl**. They are distinct concepts. **IsDeviceControl** is a boolean property available on every AccessType row, flagging whether the row drives MQTT SparkplugB device-control semantics. **DeviceControl** is simply the Name of the predefined row that ships with **IsDeviceControl=true**, used by default on MQTT SparkplugB Publisher channels. You can set **IsDeviceControl=true** on any custom AccessType you create.

Common AccessType Patterns

Continuous Monitoring

```
Name: AlwaysRead
ReadPolling: Always
ReadPollingRate: 1000
WriteEvents: False
ReadOnStartup: True
```

Control Points

```
Name: ReadWrite
ReadPolling: Always
ReadPollingRate: 500
WriteEvents: True
WriteEvent: OnChange
```

Event-Driven

```
Name: Triggered
ReadPolling: Never
ReadTrigger: @Tag.ReadNow
WriteEvents: True
WriteTrigger: @Tag.WriteNow
```

OnDisplayOrServer Mode

Optimizes communication by activating points only when needed.

How It Works

OnDisplay Behavior:

- Points activate when display opens
- Deactivate when display closes
- Reduces unnecessary communication
- Improves system performance

OnServer Behavior:

- Always active when OPC Server running
- Required for server-side logic

- Scripts, alarms, expressions need this

Configuration

1. Set **ReadPolling** to **OnDisplayOrServer**
2. Configure **ReadPollingRate**
3. Enable **WriteEvents** if needed
4. Assign to relevant points

Use Cases

Display-Driven:

```
Line A Display  Line A Tags Active
Line B Display  Line B Tags Active
No Display  No Communication
```

Server Requirements:

- Script Tasks
- Class Methods
- Expressions
- Alarm Conditions

<ac:structured-macro ac:name="info"> ac:rich-text-body Server-side tags (used in scripts, alarms, expressions) require an AccessType with ReadPolling set to "Always" or a separate AccessType for OnDisplayOrServer mode. </ac:rich-text-body> </ac:structured-macro>

Trigger-Based Operations

Read Triggers

csharp

```
// Trigger read when button pressed
ReadTrigger: @Tag.RefreshButton

// Periodic trigger
ReadTrigger: @Tag.Timer.SecondTick

// Conditional trigger
ReadTrigger: @Tag.ProductionActive && @Tag.ShiftRunning
```

Write Triggers

csharp

```
// Write on command
WriteTrigger: @Tag.SendCommand

// Batch write
WriteTrigger: @Tag.BatchReady

// Conditional write
WriteTrigger: @Tag.Value > 100
```

Performance Optimization

Polling Rate Guidelines

Data Type	Suggested Rate	Use Case
Critical Alarms	100-500ms	Safety systems
Process Values	500-1000ms	Normal monitoring
Status/States	1000-5000ms	Equipment status
Reports	5000-60000ms	Historical data

Block Commands

For protocols supporting block transfers:

```
BlockCommand: "READ_BLOCK:100:50"
```

Benefits:

- Reduced overhead
- Faster updates
- Lower network traffic
- Improved efficiency

Check driver documentation for supported commands.

Best Practices Checklist

- [ ] **Group Similar Points** - Same polling requirements
- [ ] **Use OnDisplayOrServer** - For HMI-only data
- [ ] **Minimize Always Polling** - Only critical points
- [ ] **Set Appropriate Rates** - Match process dynamics
- [ ] **Use Triggers Wisely** - Reduce continuous polling
- [ ] **Configure Startup Reads** - Initialize important values
- [ ] **Test Performance** - Monitor CPU and network load

Troubleshooting

Points not updating:

- Verify ReadPolling enabled
- Check polling rate
- Confirm display open (OnDisplay)
- Review trigger conditions

Write failures:

- Confirm WriteEvents enabled
- Check WriteEvent setting
- Verify trigger firing
- Review point access rights

Performance issues:

- Reduce polling rates
- Use OnDisplayOrServer
- Implement triggers
- Enable block commands

OnDisplay not working:

- Verify display contains tags
- Check AccessType assignment
- Confirm display runtime active
- Review server-side usage

Examples by Industry

Manufacturing

```
HMI Points: OnDisplayOrServer, 500ms
Alarms: Always, 250ms
Reports: Triggered, OnDemand
```

Building Automation

```
HVAC: Always, 5000ms
Lighting: OnDisplayOrServer, 1000ms
Energy: Always, 60000ms
```

Process Control

```
Critical: Always, 100ms
Process: Always, 1000ms
Historical: Triggered, 5000ms
```

In this section...
