# Tag Access Code

Read and write tag values via scripts.



---

Tag Access Code

- Reading and Writing Tags
- Tag Properties and Methods

Reading and Writing Tags
Tag Properties and Methods

---

## Reading and Writing Tags

The `@Tag.` prefix gives direct access to any tag in the Unified Namespace. Tag names use dot notation (`@Tag.Folder.TagName`). For tags whose name contains a `/`, see the Python note below.

**Important:** Digital tags accept the numeric values `0` and `1`, never `true` / `false`. The tag's `.State` member is a read-only boolean convenience accessor — its setter is internal and cannot be written from user scripts.

### C# Tag Access

```
// Reading tags
double temperature = @Tag.Process.Temperature;
bool motorRunning = @Tag.Motor1.Running.State;
string batchID = @Tag.Batch.ID;
DateTime timestamp = @Tag.System.CurrentTime;

// Writing tags
@Tag.Setpoint = 75.5;
@Tag.Pump.Start = 1;              // Digital tag: numeric 0 or 1, never true/false
@Tag.Pump.Stop = 0;
@Tag.Message = "Process started";  // Text tag: assign the value directly

// Array access
double[] values = @Tag.Data.Array;
@Tag.Data.Array[0] = 100;

// Quality and timestamp
TagQuality quality = @Tag.Process.Temperature.Quality;
DateTime lastUpdate = @Tag.Process.Temperature.Timestamp;
```

### Python Tag Access

Python scripts have no `@Tag.` preprocessor. Tag names **without** a `/` in them can use the `@Tag.` dot-notation form; tag names **with** a `/` must be accessed as strings via `TKGet` / `TKSet`.

```
# Reading tags (simple dotted paths)
temperature = @Tag.Process.Temperature
motor_running = @Tag.Motor1.Running.State
batch_id = @Tag.Batch.ID

# Writing tags
@Tag.Setpoint = 75.5
@Tag.Pump.Start = 1                # Digital tag: always numeric 0 or 1
@Tag.Message = "Process started"   # Text tag: assign the value directly

# Tags whose UNS path contains a '/' must be accessed as strings.
# Example: a tag at path "Plant/Line1/Temperature"
temp = TKGet("Tag.Plant/Line1/Temperature")
TKSet("Tag.Plant/Line1/Temperature", 75.5)

# NumPy on array-typed tags
import numpy as np
values = np.array(@Tag.Data.Array)
mean_value = np.mean(values)
@Tag.Statistics.Mean = mean_value
```

## Tag Properties and Methods

```
// Tag properties
public class TagProperties
{
    // Basic properties
    string Name { get; }
    object Value { get; set; }
    TagQuality Quality { get; }
    DateTime Timestamp { get; }

    // Extended properties
    double Min { get; set; }
    double Max { get; set; }
    string Units { get; }
    string Description { get; }

    // Methods
    void ForceValue(object value);
    void Subscribe(Action<TagChangeEvent> handler);
    void Unsubscribe();
    bool WriteWithConfirm(object value, int timeout);
}
```

---

**In this section...**

---