Script Types Code

Understand the different script types.

Script Types Code locations:

- Tasks (Scheduled Scripts)
- Classes (Reusable Libraries)
- Expressions (Inline Calculations)
- Display Scripts (UI Event Handlers)

Tasks (Scheduled Scripts)

Tasks execute on schedules or triggers:

csharp

```
// C# Task Example

        // Read current values
        double production = @Tag.Production.Counter;
        DateTime shiftStart = @Tag.Shift.StartTime;

        // Calculate metrics
        double hours = (DateTime.Now - shiftStart).TotalHours;
        double rate = count / hours;

        // Update tags
        @Tag.KPI.ProductionRate = rate;
        @Tag.KPI.LastUpdate = DateTime.Now;

        // Log to database
        @Script.Class.Util.LogToDatabase(rate);
```

Classes (Reusable Libraries)

Classes provide reusable functions:

csharp

```
// C# Class Library: EngineeringCalcs

    // Flow calculation
    public static double CalculateFlow(double dp, double k)
    {
        return k * Math.Sqrt(dp);
    }

    // Temperature conversion
    public static double CelsiusToFahrenheit(double celsius)
    {
        return (celsius * 9.0 / 5.0) + 32;
    }

    // Pressure conversion
    public static double PSIToBar(double psi)
    {
        return psi * 0.0689476;
    }

    // Tank volume
    public static double CylindricalTankVolume(double diameter, double level)
    {
        double radius = diameter / 2;
        return Math.PI * radius * radius * level;
    }
```

Expressions (Inline Calculations)

Expressions are single-line calculations:

javascript

```
// JavaScript Expressions
// Conditional logic
@Tag.Pump.Status = @Tag.Pump.Running ? "Running" : "Stopped"

// Mathematical calculation
@Tag.Tank.Volume = Math.PI * Math.pow(@Tag.Tank.Radius, 2) * @Tag.Tank.Level

// String manipulation
@Tag.Display.Message = "Temperature: " + @Tag.Temp.Value.toFixed(1) + "°C"

// Date/time calculation
@Tag.Batch.Duration = (Date.now() - @Tag.Batch.StartTime) / 1000 / 60
```

Display Scripts (UI Event Handlers)

Scripts triggered by UI events:

csharp

```
// C# Display Script
public void OnButtonClick(object sender, EventArgs e)
{
    // Validate user input
    if (!ValidatePermissions())
    {
        ShowMessage("Access Denied");
        return;
    }

    // Execute command
    @Tag.Equipment.StartCommand = true;

    // Log action
    LogUserAction("Start button pressed");

    // Update display
    RefreshDisplay();
}

private bool ValidatePermissions()
{
    return @Tag.Security.UserLevel >= 2;
}
```

In this section...
