Database Operations Code

Perform database operations from scripts.

In this page:

SQL Database Access

```
// C# Database Operations
public class DatabaseScript
{
    public void ReadFromDatabase()
    {
        // Simple query
        var result = @Dataset.Query("SELECT * FROM Production WHERE Date = TODAY");

        // Parameterized query
        var parameters = new Dictionary<string, object>
        {
            { "StartDate", DateTime.Today },
            { "EndDate", DateTime.Now }
        };

        var data = @Dataset.ExecuteQuery(
            "SELECT * FROM Production WHERE Timestamp BETWEEN @StartDate AND @EndDate",
            parameters
        );

        // Process results
        foreach (var row in data.Rows)
        {
            @Tag.Production.Count = row["Count"];
            @Tag.Production.Quality = row["Quality"];
        }
    }

    public void WriteToDatabase()
    {
        // Insert data
        @Dataset.Execute(
            "INSERT INTO BatchRecords (BatchID, StartTime, Product) VALUES (@ID, @Time, @Product)",
            new {
                ID = @Tag.Batch.ID,
                Time = DateTime.Now,
                Product = @Tag.Batch.Product
            }
        );

        // Update data
        @Dataset.Execute(
            "UPDATE Equipment SET RunHours = @Hours WHERE ID = @EquipID",
            new {
                Hours = @Tag.Motor.RunHours,
                EquipID = @Tag.Motor.ID
            }
        );
    }
}
```

Transaction Management

```
// Transaction example
public void ExecuteTransaction()
{
    using (var transaction = @Dataset.BeginTransaction())
    {
        try
        {
            // Multiple operations in transaction
            @Dataset.Execute("INSERT INTO Orders ...", transaction);
            @Dataset.Execute("UPDATE Inventory ...", transaction);
            @Dataset.Execute("INSERT INTO Audit ...", transaction);

            // Commit if all successful
            transaction.Commit();
        }
        catch (Exception ex)
        {
            // Rollback on error
            transaction.Rollback();
            @Tag.Error.Message = ex.Message;
        }
    }
}
```

In this section...
