Dataset Namespace Reference

Access dataset objects via scripts.

The Datasets Namespace exposes properties and methods from the . NET objects used by the Datasets module execution. You can use these properties and methods on your Displays or to create Scripts and Alarms.

Datasets Runtime Attributes
Monitoring Database Connection Status

Datasets Runtime Attributes

Below is a table describing some properties from the Datasets Namespace. Access the Datasets Namespace page for a complete list of available options.

Property	Type	Description	Example
`CursorIndex`	Integer	Represents the current position of a cursor within the result set of a query. It is an integer value that indicates the row number in the result set that the cursor points to, with the first row having an index of 0. The cursor can be used to navigate through the result set, allowing you to read, update, or delete rows of data.	`@Dataset.Query.QueryName.CursorIndex + 1`
`OpenStatusMessage`	String	Provides a human-readable message describing the current state of the database connection. This can be useful for monitoring the connection status, diagnosing issues, or displaying the connection status to users in a user-friendly format.	`@Dataset.DB.DatabaseName.OpenStatusMessage`
`ConnectionString`	String	Used to store the necessary information for establishing a connection to a database. It is a string value containing details such as server name, database name, authentication credentials, and other relevant parameters.	`@Dataset.DB.ProviderName.ConnectionString`

Monitoring Database Connection Status

Monitoring Database Connections is an essential aspect of maintaining a reliable operation of the solution.

That can be accomplished using the Dataset Namespace properties.

Examples (Replace YourDatabaseName and YourQueryName to object you want to test)

1. **IsStarted:** This property indicates if the dataset is started, meaning it has been initialized and connected to the database. You can use it to check if the dataset is currently running.

```
bool isStarted = @Dataset.DB.YourDatabaseName.IsStarted;
```

2. **Query Execution Time:** You can measure the execution time of a query by checking the time before and after executing the query.

```
DateTime startTime = DateTime.Now;
@Dataset.Query.YourQueryName.SelectCommand();
DateTime endTime = DateTime.Now;
TimeSpan executionTime = endTime - startTime;
```

3. **ConnectionString:** You can check the connection string used for the database to ensure it is configured correctly.
Example

```
string connectionString = @Dataset.DB.YourDatabaseName.ConnectionString;
```

In this section...