

WinCC Unified ODBC Toolset Library

Table of Contents

ODBC Toolset library	3
Disclaimer:	3
Scope of the library:.....	3
SQL Server basic assumptions:	3
ODBC Toolset library types	4
Types > UDT > sqlUDT V1.1.4.....	4
Types > Script > ODBC Toolset V1.1.4.....	5
List_DB function	6
Create_DB function.....	7
Delete_DB function.....	8
List_Table function	8
Create_Table function.....	9
Delete_Table function.....	10
List_sProc function.....	11
Create_sProc function.....	12
Delete_sProc function	13
Execute_Query function	13
ODBC Toolset library master copies	14
Data / tag structure	14
Setup panel	16
Execute panel	16
Databases panel.....	17
Tables panel	17
Stored Procedures panel	18

SQL Table viewer panel.....	18
Logger viewer panel.....	19
Infobar	19
Project engineering information	20
User administration	20
Runtime settings	20
Tag counts	21
Type script library	21
Screen ODBC Toolset.....	22
Screen Column Array	22
Revision history.....	23

ODBC Toolset library

Disclaimer:

The examples are non-committal and do not lay any claim to completeness with regard to configuration and equipment as well as any eventualities. The examples do not represent any custom-designed solutions but shall offer only support at typical tasks. You are accountable for the proper mode of the described products yourself.

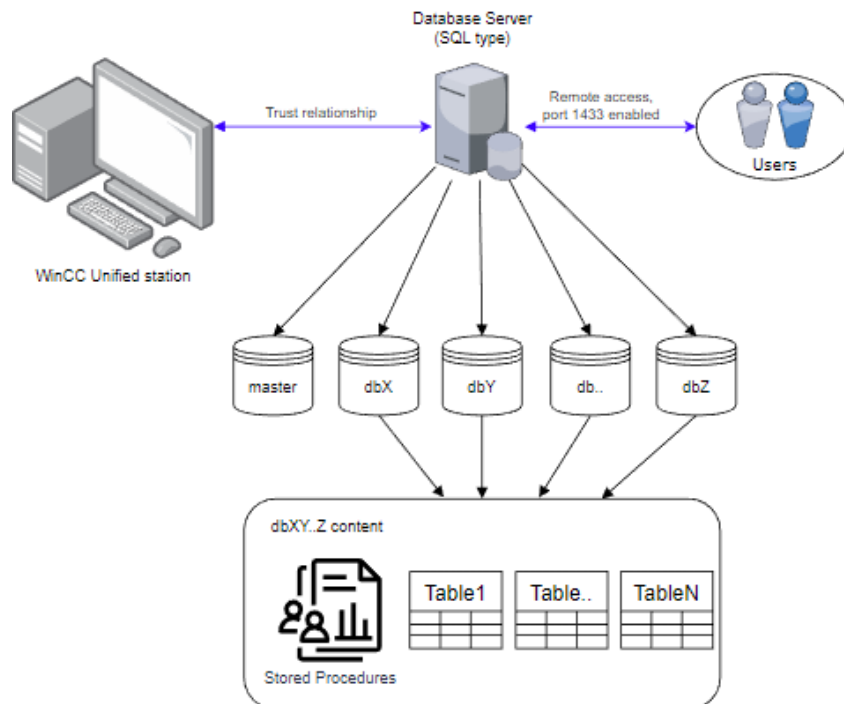
These examples do not discharge you from the obligation to safe dealing for application, installation, business and maintenance. By use of these examples, you appreciate that Siemens cannot be made liable for possible damages beyond the provisions regarding described above. We reserve us the right to carry out changes at these examples without announcement at any time. The contents of the other documentation have priority at deviations between the suggestions in these examples and other Siemens publications, such as catalogues.

Scope of the library:

The library types (script and udt) can be used to create your own SQL-type database server viewer/manager. This uses the tags in the sqlUDT, which you use as arguments in the functions that are defined in the script type library. Arguments must be provided as type string, writing down manually the tag nam. The master copies can be used to have a SQL manager already built. This library has been tested on V18 and V18.1 Unified PC Runtime and Unified Comfort Panels MTP1200, with local SQLEXPRESS and cloud Azure SQL databases, with tables up to 5 columns.

SQL Server basic assumptions:

Here is the typical SQL server architecture, which is considered by this library:

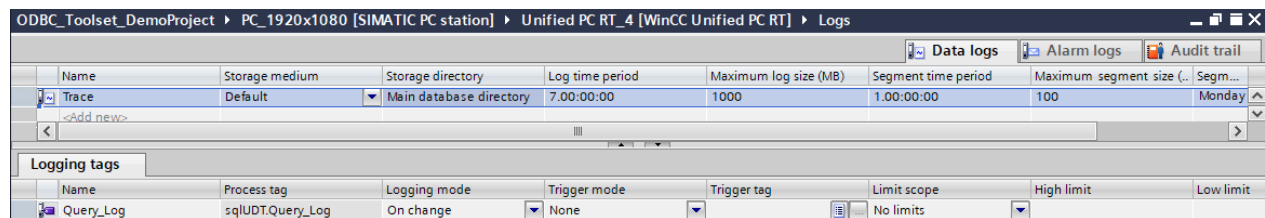


ODBC Toolset library types

Types > UDT > sqlUDT V1.1.4



This UDT must be created as an HMI Tag called “sqlUDT”, this is mandatory for the scripts function to work. A datalog called “Trace” must also be created, to have the Query_Log viewer functional.

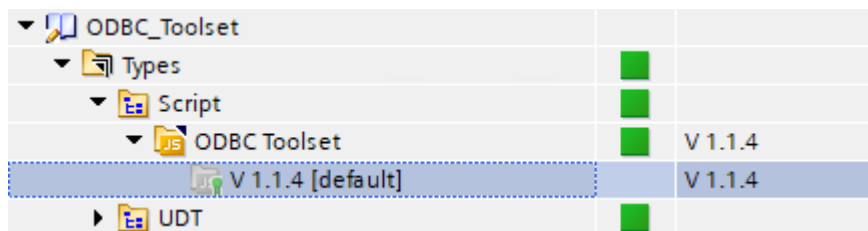


5 arrays of 100 (0 to 99) WString are also needed in the application. The following is the expected data structure to have in the Unified project:

sqlUDT	sqlUDT V 1.1.4	
endpoint	WString	SQL Server IP or DNS name, including SQL name. Eg: 192.168.0.24\SQLEXPRESS
user	WString	Username to access the server, that has remote rights.
password	WString	Password associated with the username provided.
DB_Selected	WString	The Database to work with, in the SQL Server.
DB_NameCreate	WString	Name provided to create a new DB in the SQL Server.
Table_Selected	WString	The Table to work with, in the Selected DB.
Table_NameCreate	WString	Name provided to create a new Table in the Selected DB
Table_SchemaCreate	WString	Columns provided to create a new Table in the Selected DB.
sProc_Selected	WString	Stored procedure to use in "Exec_Mode = 3" with the Selected DB.
sProc_NameCreate	WString	Name provided to create a new Stored procedure in the Selected DB
sProc_QueryCreate	WString	Store procedure "query statement" to be run with the provided NameCreate.

Exec_QueryMode	UInt	Mode 0: Free query ; Mode 1: Select ; Mode 2: Insert into ; Mode 3: Execute
Exec_QueryArguments	WString	Field to fill up the rest of the statement in regards of the mode used.
DebugInfo	WString	To display script debug info (traces) to view in the application.
Query_Log	WString	This will be logged in the Trace datalog archive, with the queryLogger function (library script)
Selected_Row	WString	This show the value of the selected row from the sqlTable viewer.
column1	Array [0..99] of WString	Column1 data – result from query
column2	Array [0..99] of WString	Column2 data – result from query
column3	Array [0..99] of WString	Column3 data – result from query
column4	Array [0..99] of WString	Column4 data – result from query
column5	Array [0..99] of WString	Column5 data – result from query

Types > Script > ODBC Toolset V1.1.4



This script contains the following functions:

ODBC Toolset	V1.1.4
Global definition	Contains internal functions: initCreds, clearEntries, tableData, queryLogger
List_DB	Sends to a List box object, the available DB in the server
Create_DB	Creates a DB in the server
Delete_DB	Delete selected DB in the server
List_Table	Sends to a List box object, the available Tables in the selected DB
Create_Table	Creates a Table in the DB
Delete_Table	Delete selected Tables in the selected DB
List_sProc	Sends to a List box object, the available stored procedures in the selected DB
Create_sProc	Creates a stored procedure in the selected DB
Delete_sProc	Delete the selected stored procedure in the selected DB

Execute_Query	Sends query to the server based on selected DB, Table or stored procedure and forward the results to a Table CWC object.
TableViewerSchema	Shape the Table CWC object, with selected table columns

List_DB function

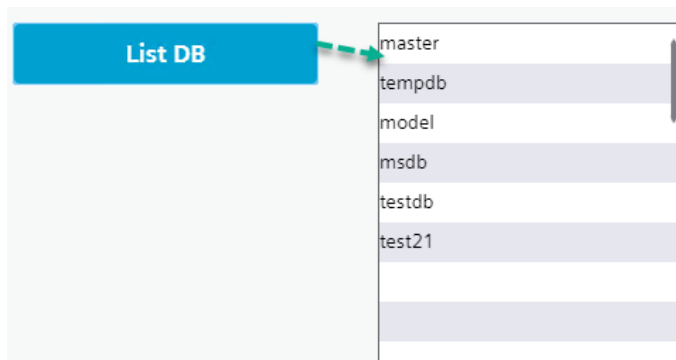
The List_DB function uses a List box object to show the results of the query: **SELECT name FROM master.sys.databases** sent to the server. Excluders can be included as arguments and they will be excluded from the result set. Selecting entry in the List box results, can write the data to sqlUDT.DB_Selected if desired.

Parameters	Arguments description
String_dbList	Mandatory, it's the List Box object name in type String.
String_Excluder_1	Optional, if you want to exclude DB's from the results, type string.
String_Excluder_2	Optional, if you want to exclude DB's from the results, type string.
String_Excluder_3	Optional, if you want to exclude DB's from the results, type string.
String_Excluder_4	Optional, if you want to exclude DB's from the results, type string.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

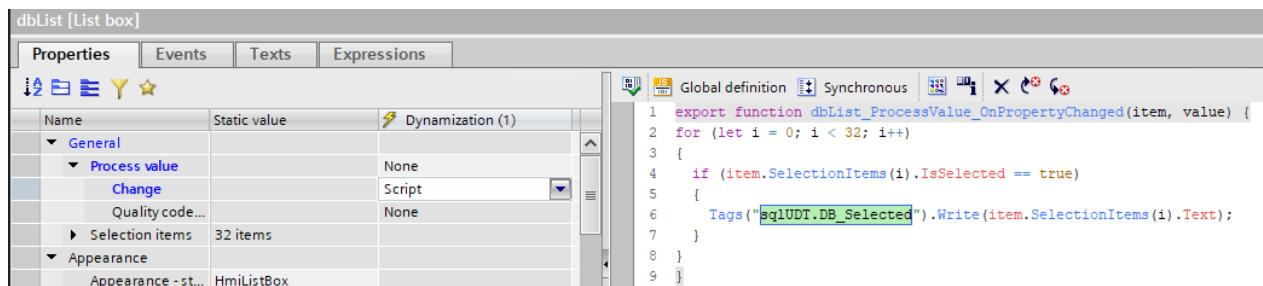
Example of values:

▼ ODBC Toolset_V_1_1_4.List_DB	
String_dbList (optional)	dbList
String_Excluder_1 (optional)	
String_Excluder_2 (optional)	
String_Excluder_3 (optional)	
String_Excluder_4 (optional)	
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:



To write the selection to the sqlUDT.DB_Selected you can implement the following on the “Process value > Change” property of the List Box:



Further notes about this function: please create 32 empty entries in the List Box object selection items.

Create_DB function

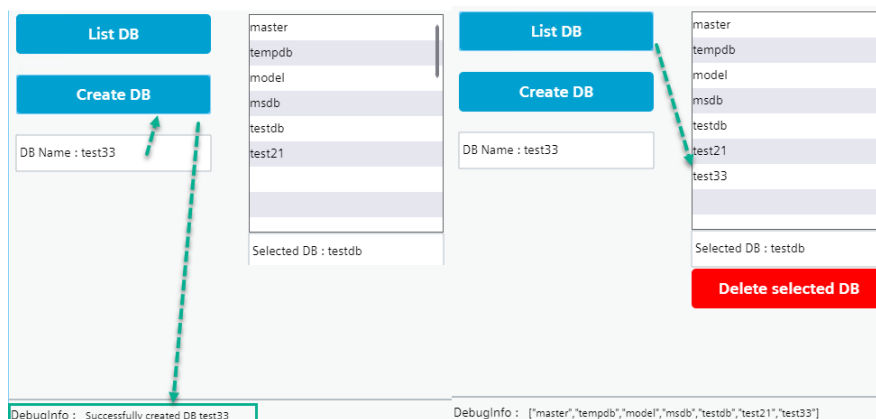
The Create_DB function will create an empty database on the server (specified endpoint), using the following query: **"CREATE DATABASE "+ dbName**. You must relist the available databases using the List_DB function afterward to see the newly created DB.

Parameters	Arguments description
String_dbName	Mandatory, it expect the tag sqlUDT.DB_NameCreate, in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.Create_DB	
String_dbName (optional)	sqlUDT.DB_NameCreate
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:



Delete_DB function

The Delete_DB function will delete the DB from the server specified by the sqlUDT.DB_Selected tag with the following query to the server: **"DROP DATABASE "+ dbName**. You must relist the entries with the List_DB function to see changes in the List Box object.

Parameters	Arguments description
String_DB_Selected	Mandatory, it expect the tag sqlUDT.DB_Selected, in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.Delete_DB	
String_DB_Selected (optional)	sqlUDT.DB_Selected
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:

The screenshot displays the ODBC Toolset_V_1_1_4.Delete_DB function interface. It features two side-by-side panels, each with a 'List DB' button, a 'Create DB' button, and a 'Delete selected DB' button. The left panel shows a list of databases: master, tempdb, model, msdb, testdb, test21, and test33. The 'test33' database is selected, and the 'Selected DB : test33' label is visible. The right panel shows the same list, but 'master' is selected, and the 'Selected DB : master' label is visible. A red arrow points from the 'Delete selected DB' button in the left panel to the 'DebugInfo' field at the bottom, which displays 'Successfully deleted DB test33'. Another red arrow points from the 'Delete selected DB' button in the right panel to the 'DebugInfo' field, which displays '['master','tempdb','model','msdb','testdb','test21']'.

List_Table function

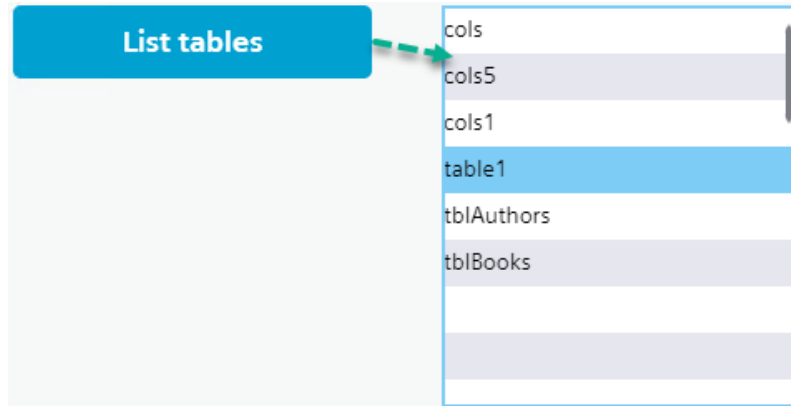
The List_Table function uses a List box object to writes the result set of the query **SELECT name FROM sys.tables** of the specified selected DB. Selecting entry in the List box results, can write the data to sqlUDT.Table_Selected if desired.

Parameters	Arguments description
String_tableList	Mandatory, it's the List Box object name in type String.
String_DB_Selected	Mandatory, it expect the tag sqlUDT.DB_Selected, in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.List_Table	
String_tableList (optional)	tableList
String_DB_Selected (optional)	sqlUDT.DB_Selected
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:



To write the selection to the sqlUDT.Table_Selected you can implement the following on the “Process value > Change” property of the List Box:

tableList [List box]

Properties Events Texts Expressions

General

Process value

Change Script

Quality code... None

Selection items 32 items

Appearance

Appearance - st... HmiListBox

```

1 export function tableList_ProcessValue_OnPropertyChanged(item, value) {
2   for (let i = 0; i < 32; i++)
3   {
4     if (item.SelectionItems(i).IsSelected == true)
5     {
6       Tags("sqlUDT.Table_Selected").Write(item.SelectionItems(i).Text);
7     }
8   }
9 }

```

Further notes about this function: please create 32 empty entries in the List Box object selection items.

Create_Table function

The Create_Table function will create a table on the selected DB, with the name and the schema (column definition), using the following query: **"CREATE TABLE " + tableName + " (" + tableColumns + ")"**. You must relist the available tables using the List_Table function afterward to see the newly created DB.

Parameters	Arguments description
String_DB_Selected	Mandatory, it expect the tag sqlUDT.DB_Selected, in type String.
String_tableName	Mandatory, it expect the tag sqlUDT.Table_NameCreate, in type String.
String_tableSchema	Mandatory, it expect the tag sqlUDT.Table_SchemaCreate, in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.Create_Table	
String_DB_Selected (optional)	sqlUDT.DB_Selected
String_tableName (optional)	sqlUDT.Table_NameCreate
String_tableSchema (optional)	sqlUDT.Table_SchemaCreate
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:

List tables

Create table

Table Name : testtable

Table Schema : name VARCHAR(50)

Selected Table : testtable

Delete selected table

List tables

Create table

Table Name : testtable

Table Schema : name VARCHAR(50)

Selected Table : testtable

Delete selected table

Delete_Table function

The Delete_Table function will delete the table from the DB specified by the sqlUDT.Table_Selected tag with the following query to the server: **"DROP TABLE "+ tableName**. You must relist the entries with the List_Table function to see changes in the List Box object.

Parameters	Arguments description
String_DB_Selected	Mandatory, it expect the tag sqlUDT.DB_Selected, in type String.
String_Table_Selected	Mandatory, it expect the tag sqlUDT.Table_Selected., in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.Delete_Table	
String_DB_Selected (optional)	sqlUDT.DB_Selected
String_Table_Selected (optional)	sqlUDT.Table_Selected
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:

List_sProc function

The List_sProc function uses a List box object to writes the result set of the query **"SELECT SCHEMA_NAME(SCHEMA_ID) AS [Schema], name FROM sys.objects WHERE type = 'P';"** of the specified selected DB. Selecting entry in the List box results, can write the data to sqlUDT.sProc_Selected if desired.

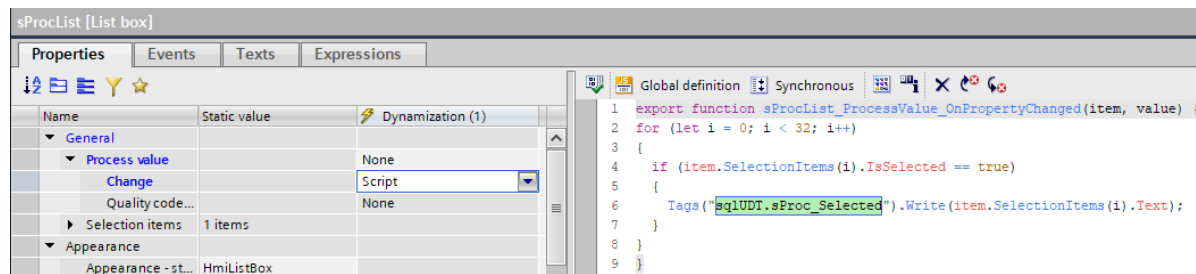
Parameters	Arguments description
String_sProcList	Mandatory, it's the List Box object name in type String.
String_DB_Selected	Mandatory, it expect the tag sqlUDT.DB_Selected, in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.List_sProc	
String_sProcList (optional)	sProcList
String_DB_Selected (optional)	sqlUDT.DB_Selected
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:

To write the selection to the sqlUDT.Table_Selected you can implement the following on the “Process value > Change” property of the List Box:



Further notes about this function: please create 32 empty entries in the List Box object selection items.

Create_sProc function

The Create_sProc function will create a stored procedure on the selected DB, with the name and the query to be stored, using the following query: **"CREATE PROCEDURE " + sProcName + " AS " + sProcQuery** . You must relist the available tables using the List_sProc function afterward to see the newly created DB.

Parameters	Arguments description
String_DB_Selected	Mandatory, it expect the tag sqlUDT.DB_Selected, in type String.
String_sProc_NameCreate	Mandatory, it expect the tag sqlUDT.sProc_NameCreate, in type String.
String_sProc_QueryCreate	Mandatory, it expect the tag sqlUDT.sProc_QueryCreate, in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.Create_sProc	
String_DB_Selected (optional)	sqlUDT.DB_Selected
String_sProc_NameCreate (optional)	sqlUDT.sProc_NameCreate
String_sProc_QueryCreate (optional)	sqlUDT.sProc_QueryCreate
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:

List stored proc.

Create stored proc.

sProc name : top1000

sProc query statement : BEGIN ...

top10

top100

Selected sProc :

Delete selected sProc.

List stored proc.

Create stored proc.

sProc name : top1000

sProc query statement : BEGIN ...

top1000

top10

top100

Selected sProc : top1000

Delete selected sProc.

Delete_sProc function

The Delete_sProc function will delete the stored procedure from the DB specified by the sqlUDT.DB_Selected tag with the following query to the server: **"DROP PROCEDURE " + selectedProc**. You must relist the entries with the List_sProc function to see changes in the List Box object.

Parameters	Arguments description
String_DB_Selected	Mandatory, it expect the tag sqlUDT.DB_Selected, in type String.
String_sProc_Selected	Mandatory, it expect the tag sqlUDT.Table_Selected., in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.Delete_sProc	
String_DB_Selected (optional)	sqlUDT.DB_Selected
String_sProc_Selected (optional)	sqlUDT.sProc_Selected
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result:

The screenshot displays two identical side-by-side panels for managing stored procedures. Each panel contains the following elements from top to bottom: a blue button labeled 'List stored proc.', a blue button labeled 'Create stored proc.', a text input field with 'sProc name : top1000', a text area with 'sProc query statement : BEGIN ...', a list box containing 'top1000', 'top10', and 'top100', a label 'Selected sProc : top1000', and a red button labeled 'Delete selected sProc.'.

Execute_Query function

The Execute_Query function will use the selected mode, where:

Modes	Arguments description
0	Free query, query = queryArg + ";;";
1	SELECT, query = "SELECT " + queryArg + " FROM " + selectedTable + ";;";
2	INSERT, query = "INSERT INTO " + selectedTable + " VALUES (" + queryArg + ")" + ";;";
3	EXECUTE, query = "EXECUTE " + selectedProc + ";;";

It's needed to complete the function with mandatory function parameters, as follows:

Parameters	Arguments description
String_Exec_QueryMode	Mandatory, it expect the tag sqlUDT.Exec_QueryMode, in type String.
String_QueryArguments	Mandatory, it expect the tag sqlUDT.Exec_QueryArguments, in type String.
String_DB_Selected	Mandatory, it expect the tag sqlUDT.DB_Selected, in type String.
String_Table_Selected	Mandatory, it expect the tag sqlUDT.Table_Selected., in type String.
String_sProc_Selected	Mandatory, it expect the tag sqlUDT.sProc_Selected., in type String.
String_TableViewer	Mandatory, it's the Table CWC object name in type String.
String_DebugInfo	Optional, name of the tag for the Debug Info to be shown in RT.

Example of values:

▼ ODBC Toolset_V_1_1_4.Execute_Query	
String_Query_Mode (optional)	sqlUDT.Exec_QueryMode
String_QueryArguments (optional)	sqlUDT.Exec_QueryArguments
String_DB_Selected (optional)	sqlUDT.DB_Selected
String_Table_Selected (optional)	sqlUDT.Table_Selected
String_sProc_Selected (optional)	sqlUDT.sProc_Selected
String_TableViewer (optional)	sqlTable
String_DebugInfo (optional)	sqlUDT.DebugInfo

Example of result (mode 1):

Execute	Setup
Mode : 1	*
Execute query	
Databases	Tables
List tables	cols
	cols5
Stored Procedures	SQL Table Viewer
	Index
	name
	age
	0
	dom
	33
	1
	dominic
	34
	2
	bob
	35

ODBC Toolset library master copies

This part will cover the master copies information and engineering.

Data / tag structure

In the demo project, there's a tag table called "sql", where you have the following variables:

▼ HMI tags
Show all tags
Add new tag table
Default tag table [8]
sql [6]
Template [17]

Tag name	Data type	Comment
sqlUDT	sqlUDT V 1.1.4	
endpoint	WString	SQL Server IP or DNS name, including SQL name. Eg: 192.168.0.24\SQLEXPRESS
user	WString	Username to access the server, that has remote rights.
password	WString	Password associated with the username provided.
DB_Selected	WString	The Database to work with, in the SQL Server.
DB_NameCreate	WString	Name provided to create a new DB in the SQL Server.
Table_Selected	WString	The Table to work with, in the Selected DB.
Table_NameCreate	WString	Name provided to create a new Table in the Selected DB
Table_SchemaCreate	WString	Columns provided to create a new Table in the Selected DB.
sProc_Selected	WString	Stored procedure to use in "Exec_Mode = 3" with the Selected DB.
sProc_NameCreate	WString	Name provided to create a new Stored procedure in the Selected DB
sProc_QueryCreate	WString	Store procedure "query statement" to be run with the provided NameCreate.
Exec_QueryMode	UInt	Mode 0: Free query ; Mode 1: Select ; Mode 2: Insert into ; Mode 3: Execute
Exec_QueryArguments	WString	Field to fill up the rest of the statement in regards of the mode used.
DebugInfo	WString	To display script debug info (traces) to view in the application.
Query_Log	WString	This will be logged in the Trace datalog archive, with the queryLogger function (library script)
Selected_Row	WString	This show the value of the selected row from the sqlTable viewer.
column1	Array [0..99] of WString	Column1 data – result from query
column2	Array [0..99] of WString	Column2 data – result from query
column3	Array [0..99] of WString	Column3 data – result from query
column4	Array [0..99] of WString	Column4 data – result from query
column5	Array [0..99] of WString	Column5 data – result from query

The following will explain how these tags are used per functionality.

Setup panel

Tags used:

sqlUDT.endpoint	SQL Server IP or DNS name, including SQL name. Eg: 192.168.0.24\SQLEXPRESS
sqlUDT.user	Username to access the server, that has remote rights.
sqlUDT.password	Password associated with the username provided.

Execute

Setup

Endpoint :

User :

Password :

Execute panel

Tags used:

sqlUDT.Exec_QueryMode	Mode 0: Free query ; Mode 1: Select ; Mode 2: Insert into ; Mode 3: Execute
sqlUDT.Exec_QueryArguments	Field to fill up the rest of the statement in regards of the mode used.

Available modes (see Examples chapter at the end):

Modes	Arguments description
0	Free query, query = queryArg + ";;";
1	SELECT, query = "SELECT " + queryArg + " FROM " + selectedTable + ";;";
2	INSERT, query = "INSERT INTO " + selectedTable + " VALUES (" + queryArg + ")" + ";;";
3	EXECUTE, query = "EXECUTE " + selectedProc + ";;";

Execute

Setup

Mode : 1

*

Execute query

Databases panel

List, create and delete DB on the server.

Databases

Tables

Stored Procedures

List DB

Create DB

DB Name : test33

master

tempdb

model

msdb

testdb

test21

Selected DB : testdb

Delete selected DB

Tables panel

List, create and delete Tables on the DB.

Databases

Tables

Stored Procedures

List tables

Create table

Table Name : testtable

Table Schema : name VARCHAR(50)

cols

cols5

cols1

tblAuthors

tblBooks

Selected Table : cols

Delete selected table

Stored Procedures panel

List, create and delete Stored Procedures on the DB.

Databases

Tables

Stored Procedures

List stored proc.

Create stored proc.

sProc name : top1000

sProc query statement : BEGIN ...

top10

top100

Selected sProc : top1000

Delete selected sProc.

SQL Table viewer panel

Result set of the Execute Query is send to this Table CWC. Sorting columns and selection of rows are possible. Also, the selected row data will be passed to the tag `sqlUDT.Selected_Row`, on the Table CWC event "SelectedRow". The selected row data is a JavaScript object, to easily return specific data from selection. In the demo project, the whole object is parsed as a string to be returned. The Index is a manual-made column by the library, it refers to the array index of the tag that is assigned. For example, `column1[1]` tag value would be "dominic" and `column2[2]` would be "35".

SQL Table Viewer

Index	name	age
0	dom	33
1	dominic	34
2	bob	35

sqlTable [Table]

Properties

Events

Texts

Expressions

Activated

Deactivated

SelectedRow

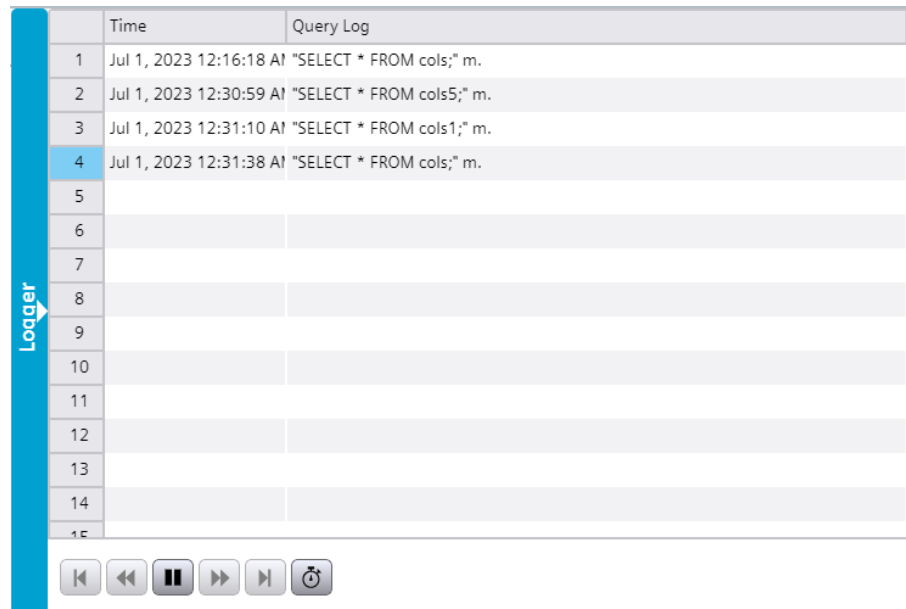
Global definition

Synchronous

```
1 export function sqlTable_OnSelectedRow(item, RowContent) {
2   Tags("sqlUDT.Selected_Row").Write(JSON.stringify(RowContent));
3 }
```

Logger viewer panel

The logger panel is a Process Control object that reads last hour of the sqlUDT.Query_Log:Query_Log of the Trace log.

The image shows a 'Logger' panel with a vertical blue bar on the left containing the word 'Logger' and a right-pointing arrow. The main area is a table with two columns: 'Time' and 'Query Log'. The table has 15 rows, numbered 1 to 15 on the left. Rows 1, 2, 3, and 4 contain data, while rows 5 through 15 are empty. Below the table is a control bar with six buttons: a left arrow, a double left arrow, a pause button, a double right arrow, a right arrow, and a refresh/clock button.

	Time	Query Log
1	Jul 1, 2023 12:16:18 AM	"SELECT * FROM cols;" m.
2	Jul 1, 2023 12:30:59 AM	"SELECT * FROM cols5;" m.
3	Jul 1, 2023 12:31:10 AM	"SELECT * FROM cols1;" m.
4	Jul 1, 2023 12:31:38 AM	"SELECT * FROM cols;" m.
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

Infobar

The infobar display two things.

1. DebugInfo, with the tag sqlUDT.DebugInfo.
2. Selected row data, with the tag sqlUDT.Selected_Row.

DebugInfo : Mode is: 1 and this is the query: SELECT Top (5) * FROM tblAuthors; Selected row data: [{"Author_name":"Author - 3","Id":"3","Index":"2","country":"Country - 3 name"}]

Project engineering information

User administration

Default user/password	
User	admin
Password	Admin123

The screenshot shows the 'Users and roles' configuration window in SIMATIC Manager. The left sidebar lists various project settings, with 'Users and roles' selected. The main area displays a table of users and a list of assigned roles.

User name	Password	Authentication ...	Runtime timeout	Comment
Anonymous				User created by default that does n.
admin	*****	Password	30 Min	PW: Admin123
<Add new user>				

Assigned to	Name	Description	Runtime timeout	Comment
<input type="checkbox"/>	Drive Administrator	System-defined role "Drive Admini...	30 Min	"Drive Administrator" role
<input type="checkbox"/>	Drive Safety Engineer	System-defined role "Drive Safety E...	30 Min	"Drive Safety Engineer" role
<input type="checkbox"/>	Drive Engineer and Service	System-defined role "Drive Enginee...	30 Min	"Drive Engineer and Service" role
<input type="checkbox"/>	Drive Operator	System-defined role "Drive Operator...	30 Min	"Drive Operator" role
<input type="checkbox"/>	Drive Guest	System-defined role "Drive Guest"	30 Min	"Drive Guest" role
<input type="checkbox"/>	Drive Ext. Role Fieldbus	System-defined role "Drive Ext. Rol...	30 Min	Anyone can change drive data via ...
<input type="checkbox"/>	Drive Ext. Role SDI Standard/Adv	System-defined role "Drive Ext. Rol...	30 Min	A user with this role can change dr...
<input checked="" type="checkbox"/>	HMI Administrator	System-defined role "HMI Administ...	30 Min	User Management, Monitor, Opera...

Runtime settings

Make sure that at least, you activated the Storage systems:

- Tag persistency
- Tag logging

The screenshot shows the 'Storage system' configuration window in SIMATIC Manager. The left sidebar lists various project settings, with 'Storage system' selected. The main area displays configuration options for the database type and locations for tag persistency, tag logging, and alarm logging.

Storage system _____

Database type

Database type: SQLite

Database location for tag persistency

Storage medium: Project folder
Folder: Folder of the runtime project

Main database location for tag logging

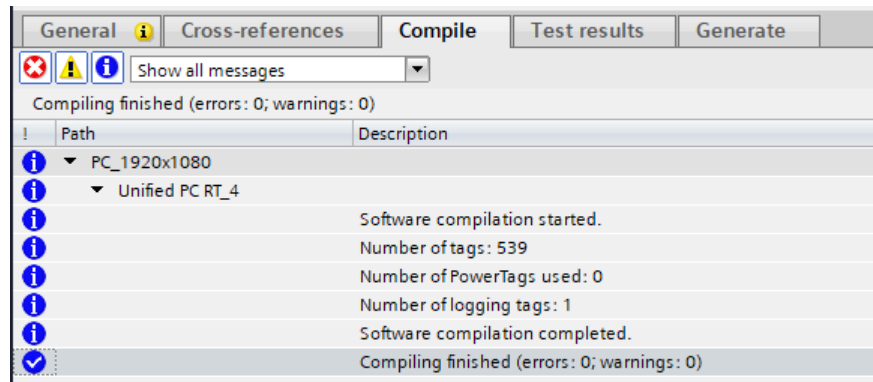
Storage medium: Project folder
Folder: Folder of the runtime project

Main database location for alarm logging

Storage medium: Project folder
Folder: Folder of the runtime project

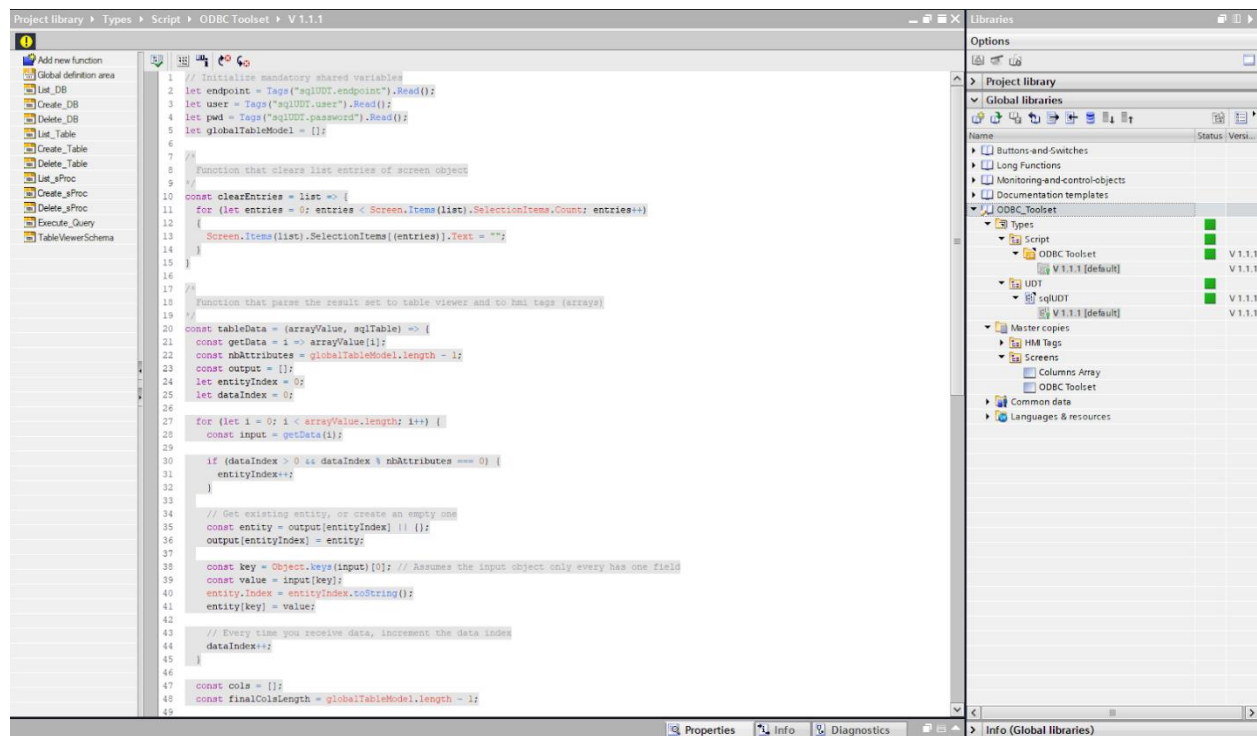
Tag counts

The whole application example will use 539 tags and 1 logging tag.



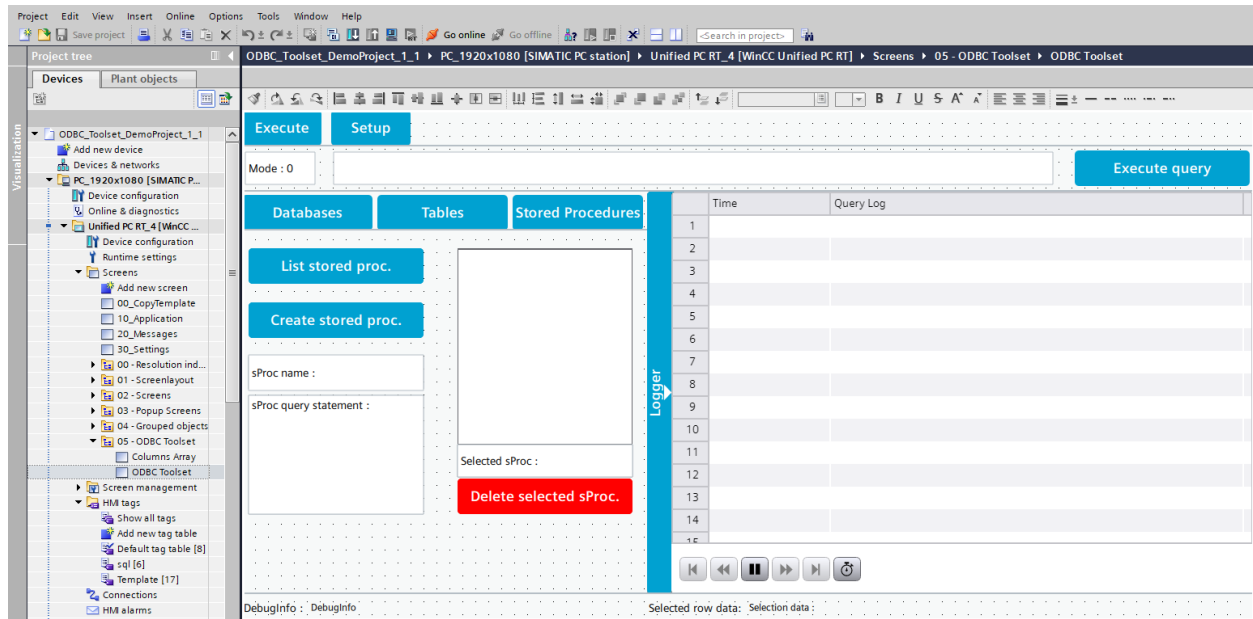
Type script library

The whole library doesn't have any know-how protection, it's freely editable.



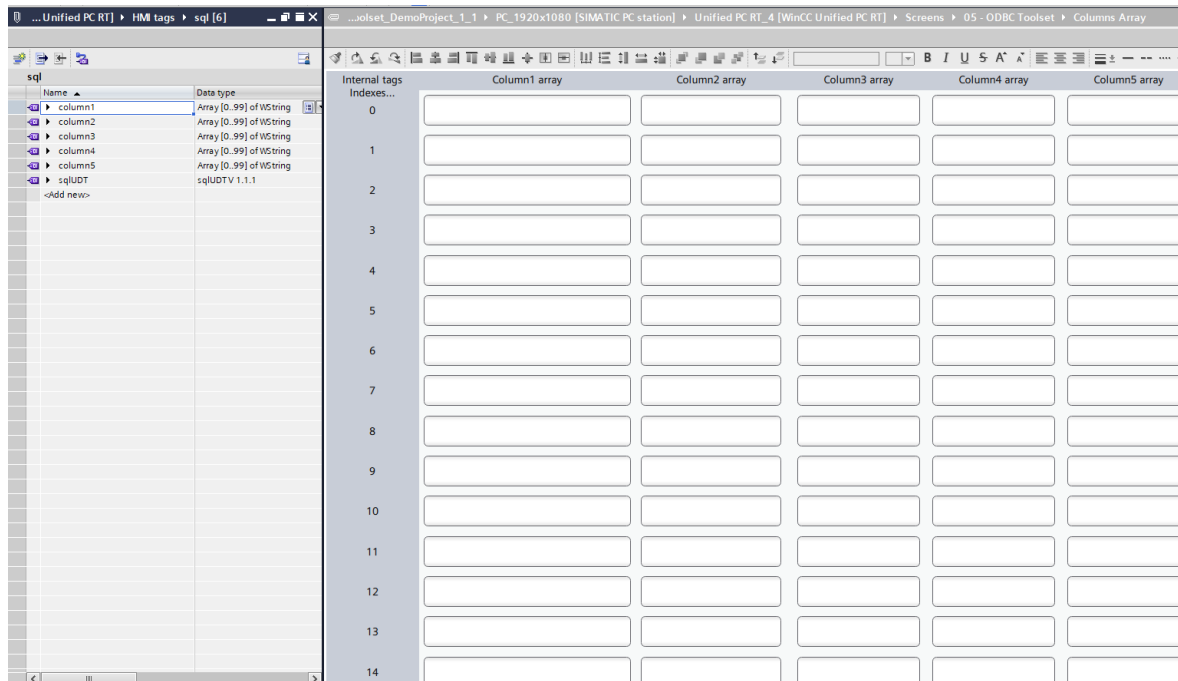
Screen ODBC Toolset

This screen is mandatory to have if you want to use the master copies. It's a design based on the HMI Template Suite, which provides intuitive design to manage a SQL server.



Screen Column Array

This screen is optional. It's only a set of I/O Fields to display data from the result set of the query sent to the DB. It represents the 5 arrays of 100 elements column1-5 in a screen.



Revision history

Here's a quick recap of the revision history of the ODBC Toolset.

21/07/23	V1.1	Added stored procedures list/create/delete/execute. Standardized all the function into a Script type. Standardized the tags into a UDT type. Created a new demo project with enhanced UI. Added master copies (screens and tag table).
15/04/23	V1.0	Public release demo project of ODBC Toolset.