

TriSys Version 8.00 ActiveX Automation Interface



Garry Lowther

Chief Software Architect

TriSys Business Software
Trinity House
Cambridge Business Park
Cambridge
CB4 0WZ
01223 393 519
www.trisys.biz

CONFIDENTIAL – INTERNAL ONLY



Table of Contents

1. INTRODUCTION	3
2. BACKGROUND	3
3. GOALS OF VERSION 8.0 AUTOMATION.....	3
4. CAUTOMATION	3
5. TRISYS OBJECT MODEL	4
6. CODE SAMPLES.....	6

1. Introduction

This working document describes the ActiveX automation interface functionality introduced in TriSys Version 8.0.

The history behind this is described as well as the interface as it stands and how to use it.

The intended audience is all staff at TriSys and technical customers who wish to extend the functionality through programming.

2. Background

TriSys was turned into an ActiveX automation server during Verison 7 development in order to allow the then scripting engine to be turned into a fully compiled business logic dynamic link library with full intellisense and debugging.

Before that time, numerous kludges were implenented to make TriSys operate as a server such as Stress testing and the Outlook Automation object to allow the Outlook COM Add-In to automate TriSys.

The redevelopment of the Office Add-Ins for greater functionality in V8 led to the development of the new programmatic interface described below.

3. Goals of Version 8.0 Automation

The principal goals are as follows:

- Allow access to the full TriSys programmable objects from .Net technology
- Expose a simple interface which could be both early and late bound
- Abstract complexity from client software
- Replace legacy utilities with fully working and refreshed utilities
- Remove legacy dependencies
- Fully document the interface

4. CAutomation

This is the new GlobalMultiUse class which can be instantiated either as an early bound object using the New keyword or late bound using the CreateObject COM function.

When instantiated, it contains properties exposing the entire TriSys object model.

5. TriSys Object Model

The object model exposes the following objects:

Object.Class	Description
TriSys8.CWordProcessor	Word processor functionality
TriSys8.CUsers	Collection of Users
TriSys8.CUserConfiguration	Configuration of each user settings
TriSys8.CUserAccessPermission	Access permissions for each user
TriSys8.CUser	User object
TriSys8.CTriSysEMailGateway	Access to Outlook object model
TriSys8.CTranslateLanguage	Translate between English and other languages
TriSys8.CTasks	Task collection object
TriSys8.CTask	Individual task object
TriSys8.CTabManager	Manage the UI tabs
TriSys8.CStrings	General purpose string functions
TriSys8.CStressTesting	Unit testing support
TriSys8.CSkills	Skill collection object
TriSys8.CSkillCategory	Skill category object
TriSys8.CSkillCategories	Skill category collection object
TriSys8.CSkill	Skill object
TriSys8.CShowForm	Dynamic forms handler
TriSys8.CShellExecute	Execute shell functions
TriSys8.CSelectSQL	Selection dialogues based on SQL Statements
TriSys8.CSelection	General purpose modal selection
TriSys8.CScreenCapture	Capture screen images
TriSys8.CReports	Reports collection object
TriSys8.CReportProperties	Report object
TriSys8.CRegistry	Windows registry CRUD
TriSys8.CRDPVirtualChannel	RDP communication channel management
TriSys8.CQueryPropertiesUser	Query properties per user
TriSys8.CQueryProperties	Query properties
TriSys8.CQueryMenuProperties	How to access queries from menus
TriSys8.CQueryMenu	Where menus to fire queries are located
TriSys8.CProgressWait	Display animated dialogues for long running tasks
TriSys8.CPrinters	Collection of printers
TriSys8.CPrinter	Individual printer object
TriSys8.CPostCodeLookupDatabase	Lookup address from a post code
TriSys8.CNumbers	Numerical functions
TriSys8.CMiscellaneous	Various not categorised functions
TriSys8.CMessageBroadcastClient	Send alerts to all subscribers
TriSys8.CMessageBox	Prompt user for yes, no, cancel etc..
TriSys8.CMergeField	Mail merge field object
TriSys8.CMailMerge	Mail merge manager
TriSys8.CLicense	License object
TriSys8.CInitialisation	Information which is determined at system initialisation
TriSys8.CIndustry	Industry object

TriSys8.CIndustries	Industry collection object
TriSys8.CHTMLEMailTemplate	Action E-Mail template management
TriSys8.CGridGammaSupport	Abstraction of legacy gamma grid for timesheets
TriSys8.CGridGammaColumns	Column collection for legacy gamma grid
TriSys8.CGridGammaColumn	Column object for legacy gamma grid
TriSys8.CGridColumnFooterInfo	Used to provide mathematical grid footers
TriSys8.CGlobalData	Standing data manager
TriSys8.CFormTabs	Form tabs collection object
TriSys8.CFormTab	Form tab
TriSys8.CForms	Collection of forms
TriSys8.CFormProperties	Individual form properties
TriSys8.CFormFields	Form field collection object
TriSys8.CFormField	Individual form field
TriSys8.CFormDesign	Form design management
TriSys8.CFormButtons	Form button collection object
TriSys8.CFormButton	Individual form button
TriSys8.CFileSystem	File system functions
TriSys8.CFileExtensions	File extensions used in CVAR
TriSys8.CFieldTypes	Field type collection object
TriSys8.CFieldType	Individual field type
TriSys8.CFieldLookups	Field lookup collection object
TriSys8.CFieldLookup	Individual field lookup
TriSys8.CFieldDescriptions	Field description collection object
TriSys8.CFieldDescription	Individual field description
TriSys8.CError	Error handling
TriSys8.CEntitySkill	Entity skill object
TriSys8.CEntitySearchFunctionParam	Parameters for entity searching
TriSys8.CEntitySearchFunction	Functions used in entity searching
TriSys8.CEntitySearchField	Entity search fields
TriSys8.CEntitySearchCriteria	Entity search criteria
TriSys8.CEntitySearch	Entity search
TriSys8.CEntityDescriptions	Entity description collection object
TriSys8.CEntityDescription	Individual entity description
TriSys8.CDiag	Diagnostics manager
TriSys8.CDBGet	Legacy database retrieval functions
TriSys8.CDatabase	TriSys database manager
TriSys8.CCrystal	Legacy crystal reports
TriSys8.CContactUsers	Users linked to contacts
TriSys8.CContacts	Contact collection object
TriSys8.CContact	Contact object
TriSys8.CConfigValue	Configuration field value
TriSys8.CCompany	Company object
TriSys8.CCompanies	Company collection object
TriSys8.CCodeJockCombo	Wrapper for legacy field types
TriSys8.CChartFXAsync	Asynchronous charting manager
TriSys8.CChartFX	Chart manager

TriSys8.CCallFusionIntegration	Telephony web service manager
TriSys8.CCachedMergeData	Mail merge data cache
TriSys8.CBigInput	Generic simple data capture
TriSys8.CAutomation	Interface to all ActiveX automation objects
TriSys8.CAnimateWait	Animation dialogue for long running tasks
TriSys8.CAddresses	Address collection object
TriSys8.CAddress	Address object
TriSys8.CActionFunctions	Action function collection object
TriSys8.CActionFunction	Action function object
TriSys8.CAction	Action object

6. Code Samples

This section gives simple examples of using the interface.

VB.Net

```
Private m_TriSysAutomation As TriSys8.CAutomation

Private Sub MyExample()

    Try

        m_TriSysAutomation = New TriSys8.CAutomation

        With m_TriSysAutomation.InitialisationInstance.User
            lblLogin.Text = .Name & " (" & .LoginName & "), Tel. " & _
                .TelNo & " is " & IIf(.SystemAdministrator, "", "not ") & _
                "a system administrator."
        End With

        Dim rs As Object = _
            m_TriSysAutomation.DatabaseInstance.OpenRecordset( _
                "Select * from Users Where LoggedIn = 1")

        Do While Not rs.EOF
            lstUsers.Items.Add(rs.fields("Name").Value)
            rs.MoveNext()
        Loop
        rs.Close()

    Catch ex As Exception

        MsgBox(ex.ToString)

    End Try
End Sub
```

C#

```

1. private TriSys8.CAutomation m_TriSysAutomation;
2.
3. private void MyExample()
4. {
5.
6.     try {
7.         m_TriSysAutomation = new TriSys8.CAutomation();
8.
9.         {
10.            lblLogin.Text = m_TriSysAutomation.InitialisationInstance.User.Name
                + " (" + m_TriSysAutomation.InitialisationInstance.User.LoginName +
                "), Tel. " + m_TriSysAutomation.InitialisationInstance.User.TelNo +
                " is " +
                (m_TriSysAutomation.InitialisationInstance.User.SystemAdministrator
                ? "" : "not ") + "a system administrator.";
11.        }
12.
13.        object rs = m_TriSysAutomation.DatabaseInstance.OpenRecordset(
                "Select * from Users Where LoggedIn = 1");
14.
15.        while (!rs.EOF) {
16.            lstUsers.Items.Add(rs.fields("Name").Value);
17.            rs.MoveNext();
18.        }
19.
20.        rs.Close();
21.    }
22.    catch (Exception ex) {
23.
24.
25.        Interaction.MsgBox(ex.ToString);
26.    }
27. }

```

Visual Basic 6

```
Dim myAutomation As TriSys8.CAutomation, sText As String
```

```
'Set myAutomation = CreateObject("TriSys8.CAutomation")
Set myAutomation = New TriSys8.CAutomation
```

```
Dim myInitialisation As TriSys8.CInitialisation
Set myInitialisation = myAutomation.InitialisationInstance
```

```
If Not myInitialisation Is Nothing Then
```

```
    sText = "Logged in as user: " & myInitialisation.User.LoginName
```

```
    Dim trisysDatabase As TriSys8.CDatabase
    Set trisysDatabase = myAutomation.DatabaseInstance
```

```
    sText = sText & vbCrLf & "Database: " & trisysDatabase.GetServer & "." & _
        trisysDatabase.GetSnapshotField("Select Db_Name()", 0)
```

```
    ' Load the contacts Grid form.
```

```
    myAutomation.MainBinderForm.StressInvocation "Main.ViewContacts", _
        "My Contacts"
```

```
End If
```