Tech Note 931
OLE Automation Examples for the QIAnalyst Chart Control

Topic#: 002747
Created: February 2013

Introduction

This *Tech Note* contains examples for automating the QIAnalyst Chart Control

Application Versions

- QI Analyst 8.0 and later

Prerequisites

- QI Analyst and InTouch must be installed on the same machine.
- At minimum, a QI Workstation license and InTouch development license should be installed on that machine.
- You should be familiar with QI Analyst OLE automation inside InTouch. (For more information refer to QI analyst OLE automation Guide and QIA80 Sample InTouch App from the Installation CD)

Adding a Control Move to a Selected Chart Point

```
{NOTE: manually select a point on Chart1 before executing the following script}
SelectedChartPoint = #Chart1.SelectedPoint; {where SelectedChartPoint is a memory integer tag)
#Chart1.AssignChartAttribute(SelectedChartPoint, 4, "Add control Move", 1);
```

Adding a Comment to a Selected Chart Point

Note: A similar example appears in the QIA OLE automation guide on Page 64.

```
{NOTE: manually select a point on Chart1 before executing the following script}
SelectedChartPoint = #Chart1.SelectedPoint; {where SelectedChartPoint is a memory integer tag)
#Chart1.AssignChartAttribute(SelectedChartPoint, 5, "This is a comment", 1);
```

Disabling a Selected Chart Point

```
{NOTE: manually select a point on Chart1 before executing the following script}

SelectedChartPoint = #Chart1.SelectedPoint; {where SelectedChartPoint is a memory integer tag)
```

```
#Chart1.AssignChartAttribute(SelectedChartPoint, 2, "Point disabled", 0);
```

Searching for a Chart Point with Value 0 and Disabling It

```
TotalPoints = #Chart1.GetValue(1001); {1001 corresponds to total points on that chart. QIA OLE automation guide Page36)
StartPoint = 1; {StartPoint : Memory Integer}

IF TotalPoints >= 0 THEN

   FOR StartPoint = 1 TO TotalPoints
      %StatPoint = #Chart1.GetStatPoint(StartPoint -1);
      PointValue = %StatPoint.Value; { PointValue: memory real)


      IF PointValue == 0 THEN

      #Chart1.AssignChartAttribute(StartPoint, 2, "disabling this point", 0);

      LogMessage(" The disabled chart point is ="+ StringFromIntg( StartPoint, 10 ));

   ENDIF;

NEXT;

ENDIF;
```

Disabling a Chart Point with a Specific Cause Description

```
TotalPoints = #Chart1.GetValue(1001); {1001 corresponds to total points on that chart. QIA OLE automation guide Page36}
StartPoint = 1; {StartPoint : Memory Integer}

IF TotalPoints >= 0 THEN


   FOR StartPoint = 1 TO (TotalPoints)

      %StatPoint = #Chart1.GetStatPoint(StartPoint -1);

      PointCauseValue = %StatPoint.Cause; { PointCauseValue : Memory Message}



      IF PointCauseValue == CauseDescription THEN {where CauseDescription : Memory Message}

         #Chart1.AssignChartAttribute(StartPoint, 2, "disabling this point", 0);

         LogMessage("The disabled chart point is ="+ StringFromIntg( StartPoint, 10 ));

      ENDIF;

   NEXT;

ENDIF;
```

Getting Cause, Action, PointValue, Createtime and List of Rule Violations from the Latest Chart Point (Last Point)

```
TotalRows = #Chart1.GetValue(1001); {TotalRows ; Memory Integer ; 1001: qisTotal}

IF TotalRows >= 0 THEN

      {Get the stat object for the last point}
      %StatPointLast = #Chart1.GetStatPoint(TotalRows);

      {Point stat for a chart point}{Before doing the below step, set the chart X-Axis label to display CreateTime}
      LastXaxisLabel = %StatPointLast.XAxisLabel; {LabelStringLast: Memory Message tag. This gets the CreateTime of the last point on chart}

      LastPointValue = %StatPointLast.Value; { LastPointValue: Memory Real}
      LastPointCause = %StatPointLast.Cause; { LastPointCause: Memory Message}
      LastPointAction = %StatPointLast.Action; { LastPointAction : Memory Message }
      LastPointRuleViolations = %StatPointLast.NRuleViolations;

      {LastPointRuleViolations: Memory Integer. This gets the number of rule violation for the last point on the chart;
      For example: If this value if "0" then it means there were no rule violations and hence the point is in control
      If this value is "3" then it means there were three rules violations and hence the point is out of control}

      {The following loop is required to evaluate each rule in the array for a chart point. These scripts also put the list of rule violations into a
listbox}
```

```
      wcClear ( "RuleViolationListBox" ); {clear the list box}
      FOR ArrayRow_i = 0 TO LastPointRuleViolations { ArrayRow_i : Memory Integer}

         RuleViolations = %StatPointLast.GetRuleViolation(ArrayRow_i); { RuleViolations : Memory Message }
         wcAddItem ( "RuleViolationListBox", RuleViolations ); {Add the violated rule name to listbox}

      NEXT;
ENDIF;
```

Resources

Here is a **link to a demo application** developed on InTouch version 10.1 SP3 that has the implementation of the above examples.

This demo app uses the **sampledb80.qia** process database, which is available by default in QI Analyst directory:

- **On 32 bit OS**: C:\Program Files\Wonderware\QIAnalyst8.0;
- **On 64 bit OS**: C:\Program Files (x86)\Wonderware\QIAnalyst8.0)

This is part of QI Analyst installation.

R. Herunde

Tech Notes are published occasionally by Wonderware Technical Support. Publisher: Invensys Systems, Inc., 26561 Rancho Parkway South, Lake Forest, CA 92630. There is also technical information on our software products at **Wonderware Technical Support.**

For technical support questions, send an e-mail to **wwsupport@invensys.com**.

Back to top