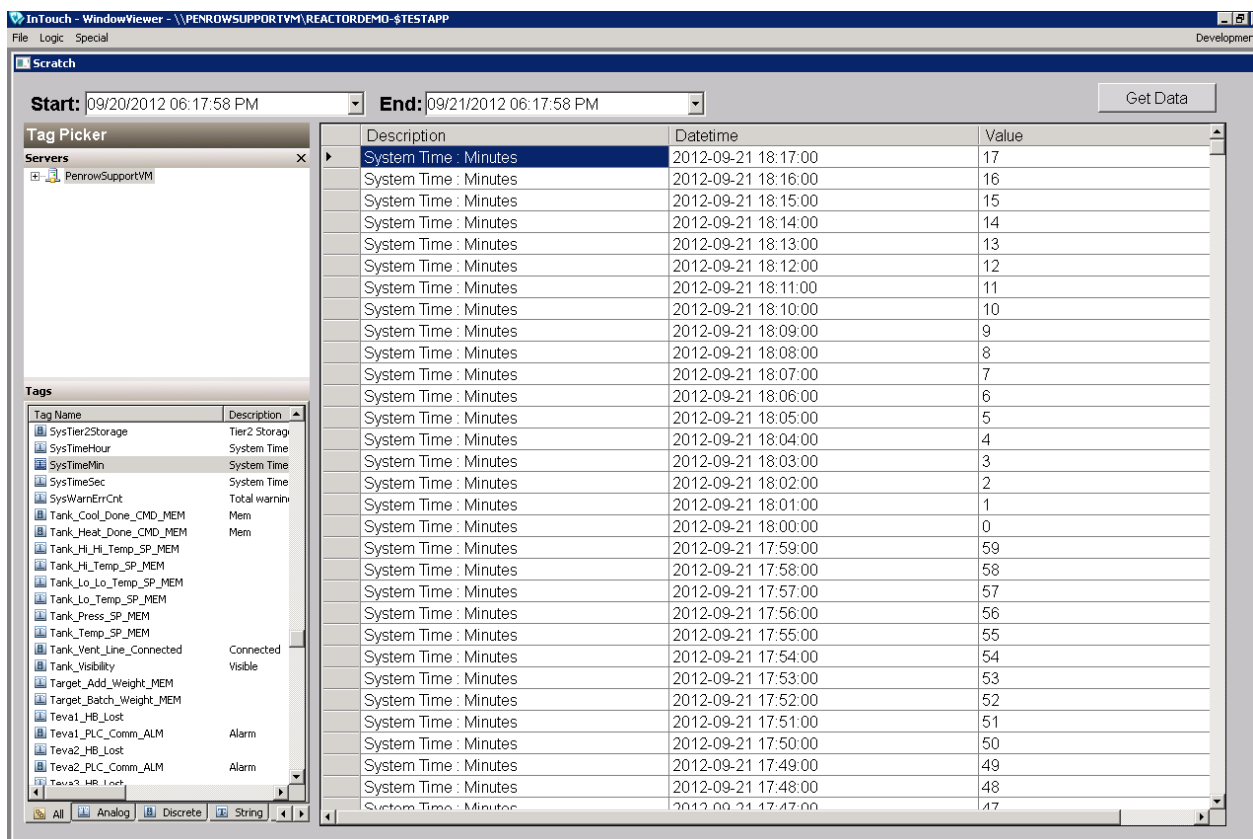


Building A Simple SQL Dataview

There is a need for a simple sql dataview inside of intouch. It is common to want to show data in an intouch screen from sql database. There are a couple prebuilt controls that come close to filling this need but fall just short enough to not be viable. The first is the SQLDatagrid archestra graphic that comes out of the box with the archestra ide. This is really intended to let intouch users add sql queries and modify sql queries from inside of InTouch, which it is good at. It is bad at being data view because it has buttons for manipulating and running queries that are unnecessary and get in the way. It is also bad because there aren't enough .net control properties exposed to control the appearance and when your goal is to present sql data appearance is critical. The other controls is the historian client query control. It does show data but presents the user with too many options to pick from. This tool is really intended for an advanced user doing analytics not just looking at a simple data screen.

The rest of this article will show you the essentials for building a sql dataview. Below is the results of what I built. My hope is that if I give you the process of how to build it you can decide to add things, modify things, or remove things. An example of something you might remove is the tag picker. I included a tagpicker so the user can decide what they want to see but maybe in your intouch application that doesn't make sense. Maybe in your application they click on a tank then a dataview with that tanks temperature and pressure and level appears.



The screenshot shows the InTouch software interface. At the top, the title bar reads "InTouch - WindowViewer - \\PENROW\SUPPORT\VM\REACTORDEMO-\$TESTAPP". Below the title bar, there are menu options: "File", "Logic", "Special", and "Development". The main window is titled "Scratch" and contains a "Tag Picker" control on the left and a data table on the right. The "Tag Picker" control has a "Servers" list with "PenrowSupportVM" selected and a "Tags" list with various tag names and descriptions. The data table has columns for "Description", "Datetime", and "Value". The table contains 17 rows of data, all with "System Time : Minutes" as the description and values ranging from 0 to 17. A "Get Data" button is located at the top right of the data table area.

Description	Datetime	Value
System Time : Minutes	2012-09-21 18:17:00	17
System Time : Minutes	2012-09-21 18:16:00	16
System Time : Minutes	2012-09-21 18:15:00	15
System Time : Minutes	2012-09-21 18:14:00	14
System Time : Minutes	2012-09-21 18:13:00	13
System Time : Minutes	2012-09-21 18:12:00	12
System Time : Minutes	2012-09-21 18:11:00	11
System Time : Minutes	2012-09-21 18:10:00	10
System Time : Minutes	2012-09-21 18:09:00	9
System Time : Minutes	2012-09-21 18:08:00	8
System Time : Minutes	2012-09-21 18:07:00	7
System Time : Minutes	2012-09-21 18:06:00	6
System Time : Minutes	2012-09-21 18:05:00	5
System Time : Minutes	2012-09-21 18:04:00	4
System Time : Minutes	2012-09-21 18:03:00	3
System Time : Minutes	2012-09-21 18:02:00	2
System Time : Minutes	2012-09-21 18:01:00	1
System Time : Minutes	2012-09-21 18:00:00	0
System Time : Minutes	2012-09-21 17:59:00	59
System Time : Minutes	2012-09-21 17:58:00	58
System Time : Minutes	2012-09-21 17:57:00	57
System Time : Minutes	2012-09-21 17:56:00	56
System Time : Minutes	2012-09-21 17:55:00	55
System Time : Minutes	2012-09-21 17:54:00	54
System Time : Minutes	2012-09-21 17:53:00	53
System Time : Minutes	2012-09-21 17:52:00	52
System Time : Minutes	2012-09-21 17:51:00	51
System Time : Minutes	2012-09-21 17:50:00	50
System Time : Minutes	2012-09-21 17:49:00	49
System Time : Minutes	2012-09-21 17:48:00	48
System Time : Minutes	2012-09-21 17:47:00	47

Building A Simple SQL Dataview

Initial Setup

Import the ActiveFactory TagPicker Control aaHistClientTagPicker.dll located at C:\Program Files (x86)\Common Files\ArchestrA. This is the .net control you see on the left of the above screenshot. This is what makes it possible for the user to pick what tag they want to see in the datagridview.

Import the script function library aaHistClientDatabase.dll from C:\Program Files (x86)\Common Files\ArchestrA. This will be necessary to make scripting work later. This essential makes this namespace available: **ArchestrA.HistClient.Database.aaServer**.

Import the .net library. This contains the datagridview. This will be the heart of displaying data. Import the script function library at C:\Windows\Microsoft.NET\Framework\v2.0.50727\System.Windows.Forms.dll

Datetime Picker Setup

The key here is to have the correct format. Open up the animations screen for the DateTime Picker then set the format to custom then set it to this: MM/dd/yyyy hh:mm:ss tt. Also don't forget to draw two tagpickers on the screen like in the screenshot.

TagPicker Setup

When the user selects a tag then needs to be a way to update the custom property "tags" so that another sql query can run to update the datagridview. This can be done by going to animations on the tag picker then select the "OnTagsSelected" even script then add the below script. Make sure to make a custom property called tags.

```
dim tag as ArchestrA.HistClient.Database.aaTag;  
tag = selectedTag(0);  
  
tags = tag.name;
```

Building the SQL Query

I created a datachange script called update that ran anytime the discrete custom property update was changed (by the way create a custom property called update). Then I put in the script below.

```
dim connection as System.Data.SqlClient.SqlConnection;  
dim command as System.Data.SqlClient.SqlCommand;  
dim reader as System.Data.SqlClient.SqlDataReader;  
LogMessage("here is a debug message 1");  
  
connection = new  
System.Data.SqlClient.SqlConnection("server=penrowsupportvm;uid=sa;database=R  
untime;password=app|topic!1024");
```

Building A Simple SQL Dataview

```
connection.Open();

LogMessage("here is a debug message 2");

DateTimePicker1.Value = System.DateTime.Now.AddDays(-1);
DateTimePicker2.Value = System.DateTime.Now;

command = new System.Data.SqlClient.SqlCommand("SET NOCOUNT ON DECLARE
@StartDate DateTime DECLARE @EndDate DateTime SET @StartDate ='" +
DateTimePicker1.Value + "' SET @EndDate = '" + DateTimePicker2.Value + "' SET
NOCOUNT OFF SELECT Description ,DateTime = convert(nvarchar, DateTime, 20)
,Round(Value,2) AS Value From ( SELECT * FROM History WHERE
History.TagName IN ('" + Tags + "') AND wwRetrievalMode = 'Delta' AND
wwVersion = 'Latest' AND DateTime >= @StartDate AND DateTime <= @EndDate)
temp LEFT JOIN Tag ON Tag.TagName =temp.TagName LEFT JOIN AnalogTag ON
AnalogTag.TagName =temp.TagName LEFT JOIN EngineeringUnit ON AnalogTag.EUKey
= EngineeringUnit.EUKey LEFT JOIN QualityMap ON QualityMap.QualityDetail =
temp.QualityDetail WHERE temp.StartDateTime >= @StartDate ORDER BY DateTime
DESC", connection);
reader = command.ExecuteReader();

LogMessage("here is a debug message 3");

dataGridView1.Rows.clear();

dataGridView1.ColumnCount = 3;
dataGridView1.Columns[0].Name = "Description";
dataGridView1.Columns[1].Name = "Datetime";
dataGridView1.Columns[2].Name = "Value";

dataGridView1.Columns[0].Width = 313;
dataGridView1.Columns[1].Width = 314;
dataGridView1.Columns[2].Width = 313;

LogMessage("Before adding the rows to the dataGridView");
while reader.Read()
    dataGridView1.Rows.add(reader("Description"), reader("DateTime"), reader(
"Value"));
endwhile;
reader.Close();
connection.Close();
LogMessage("End of the script reached");
```