Oct, 2019

Bypass Alarm Using ScadaPhone WebServer

Contents

OVERVIEW:.....3

Step 1: Configure Runtime Enable fields for all alarms 4

Step 2: Define RTE Fields in all alarms 6

Step 3: Adding ".RTE" Tags To Web Server Tag Lists: 7

Step 4: Adding Alarm Messages as Tag Descriptions.....10

Step 5: Configuring Direct Links to Specific Tag List Pages 12

OVERVIEW:

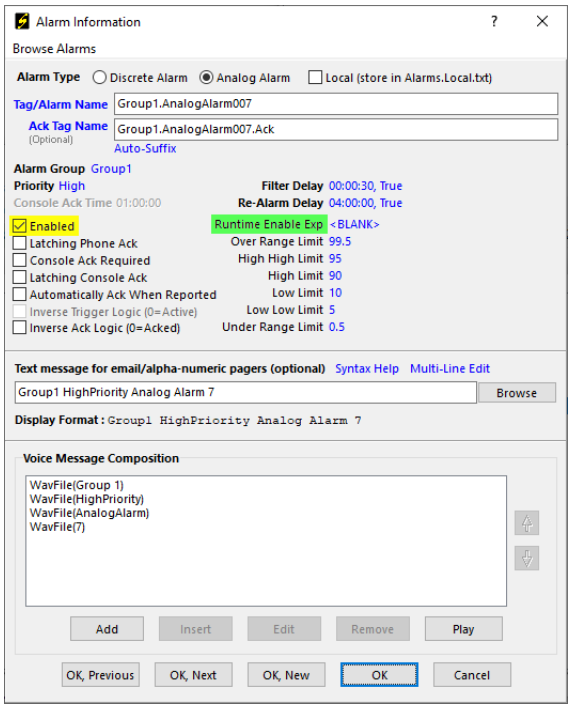
In almost every SCADA system, situations occur in which nuisance alarms are reported to operators who are well-aware of the situation being reported (e.g. a piece of equipment or a sensor is malfunctioning and is in the process of being fixed), but simply acknowledging the alarm does not always stop the repetitive reporting because the alarm signal may be "bouncing" between OFF and ON, and each bounce is interpreted as a new reportable alarm cycle.

In these situations, the system designer may opt to allow the system operators to place specific alarms into a **bypass state** to stop the nuisance alarm reporting. There are a number of ways this can be accomplished:

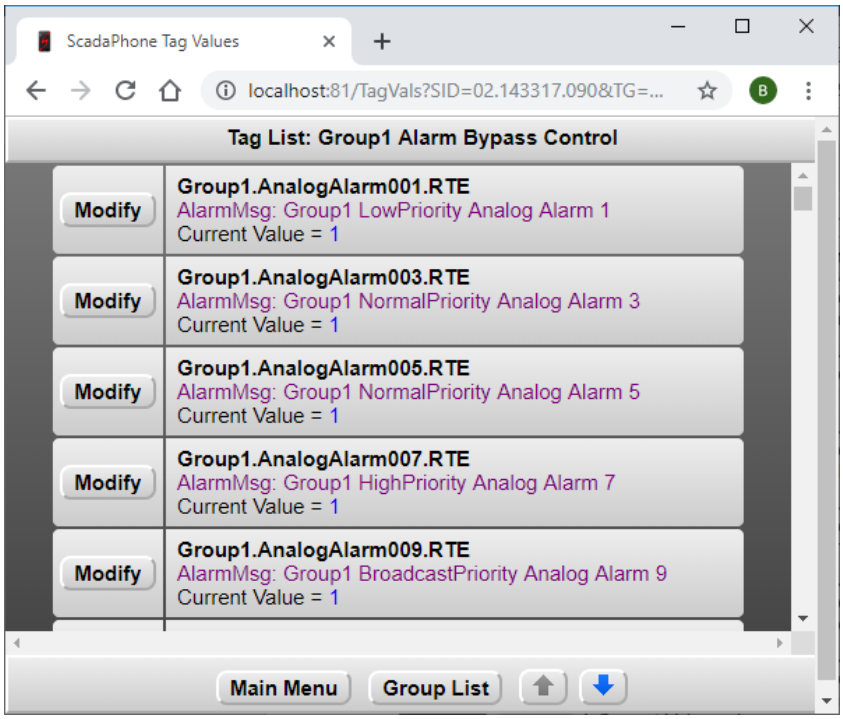
1. The alarm can be **disabled in the SCADA server**.

2. The alarm can be **disabled in ScadaPhone** using the **Enabled** field in the **Alarm Information** window, but this requires **Supervisor Access** in ScadaPhone (to switch ScadaPhone from **Run Mode** to **Development Mode**, modify the **Alarm Enable** field, and return ScadaPhone to **Run Mode**).
3. The alarm can be **disabled in ScadaPhone** using the **Runtime Enable Expression** (which is an optional field available in all ScadaPhone alarm definitions).

The third method, using the **Runtime Enable Expression**, is is covered in this application note.



The goal of this application note is the creation of an interactive table of discrete tags which can be controlled by system operators for the purpose of silencing nuisance alarms. ScadaPhone's **Web Server Interface** provides an ideal means for creating such an interactive table:



In the example above, each table entry contains three pieces of information:

1. A ScadaPhone tag name with an extension **".RTE**" (**RTE** stands for **Run Time Enable**)
2. A copy of the **Alarm Message** associated with the **RTE** tag (this helps the operator identify the correct alarm to bypass if the **RTE** tag name is cryptically abbreviated).
3. The **Current Value** of the **RTE** tag. If the **Current Value** = 1, then the associated alarm is enabled; to place the associated alarm into bypass mode, the operator can set the **Current Value** to **0** by clicking the **Modify** button.

The following pages outline the steps necessary to configure this feature.

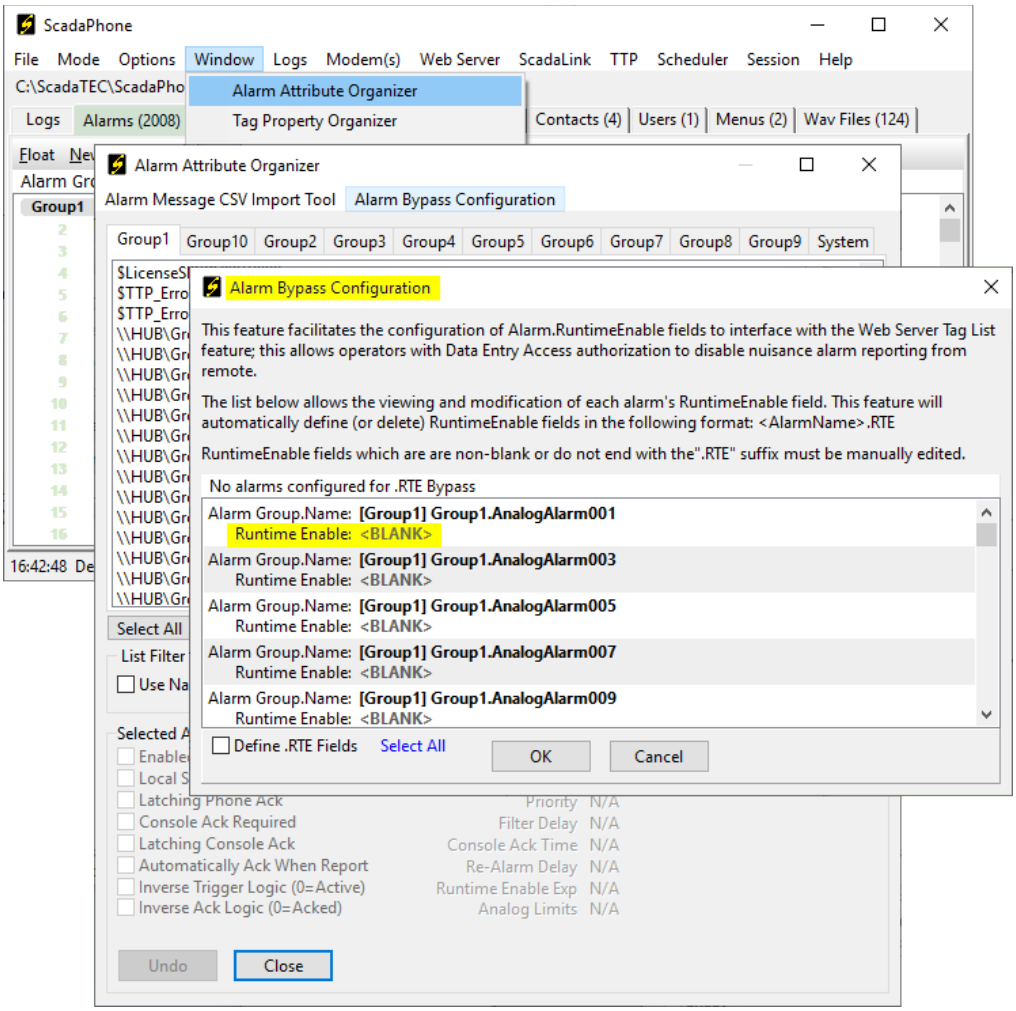
Note: This document assumes that alarms have already been configured in a ScadaPhone project. If alarms have not yet been configured, complete the alarm configuration before proceeding.

Step 1: Configure Runtime Enable fields for all alarms

As shown in the image on page 3 of this document, each ScadaPhone alarm has an optional **Runtime Enable Exp** field. This field is denoted as an "*expression*" because any valid Boolean expression (such as **$Hour > 8**) can be used to drive the **Runtime Enable** status; however, in the **Alarm Bypass** feature, each **Runtime Enable Exp** consists solely of a **Discrete Holding Tag** having the name of the **Alarm Trigger Tag** with the suffix extension "**.RTE**" added.

ScadaPhone has a configuration window specifically designed to facilitate the configuration of the **Runtime Enable Expressions** for all alarms. To open this window:

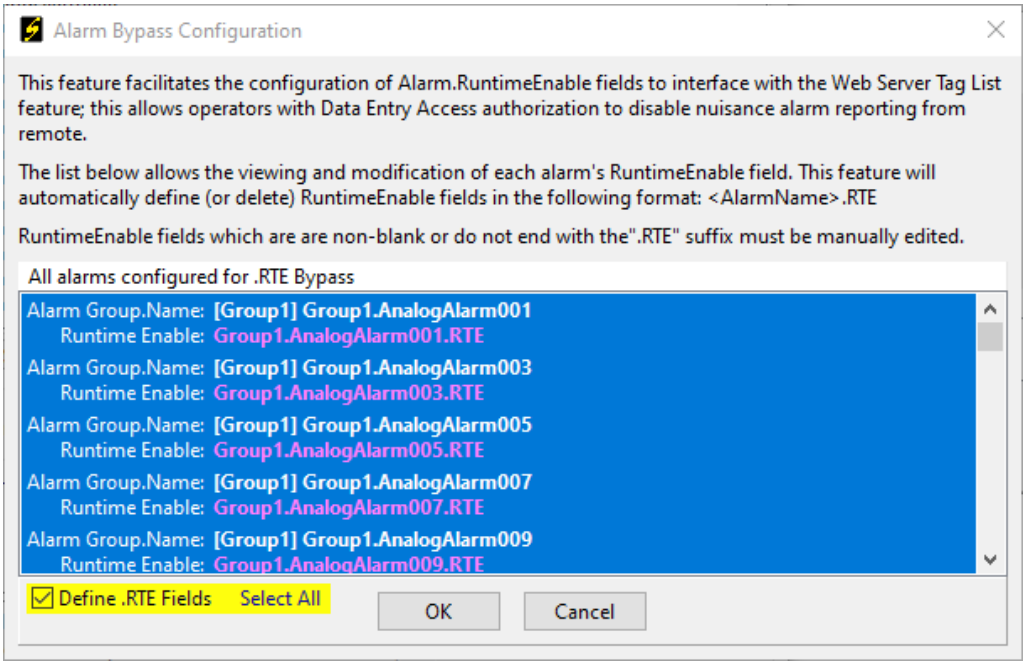
- Place ScadaPhone into **Development Mode.**
- Click the **Window** | **Alarm Attribute Organizer** menu item from ScadaPhone's main window.
- From the top menu bar of the **Alarm Attribute Organizer** window, click **Alarm Bypass Configuration.**
- The **Alarm Bypass Configuration** window shows a complete list of project alarms and their **Runtime Enable** settings. Alarms lacking a value in the optional **Runtime Enable** field will display **<BLANK>**.



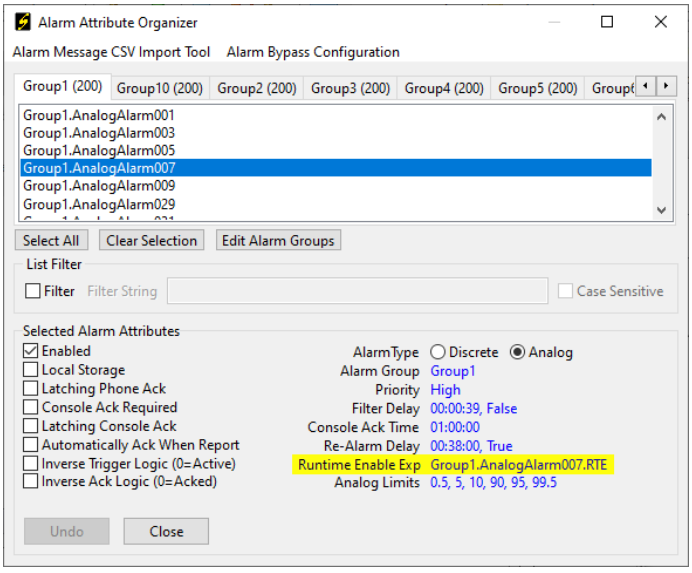
Step 2: Define RTE Fields in all alarms

The **Alarm Bypass Configuration** window allows the user to define **RTE** fields for all alarms with just 3 clicks:

- Click the **Select All** label.
- Put a check mark in the **Define .RTE Fields** box.
- Click **OK.**

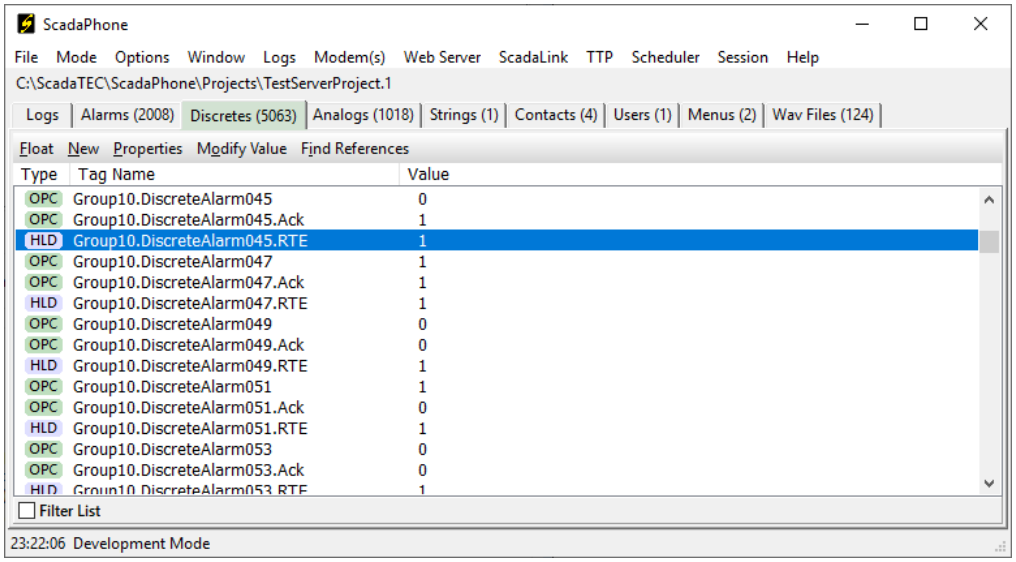


After the **Alarm Bypass Configuration** window is closed, clicking any of the alarms in the **Alarm Attribute Organizer** will display the **Runtime Enable Expression** changes:



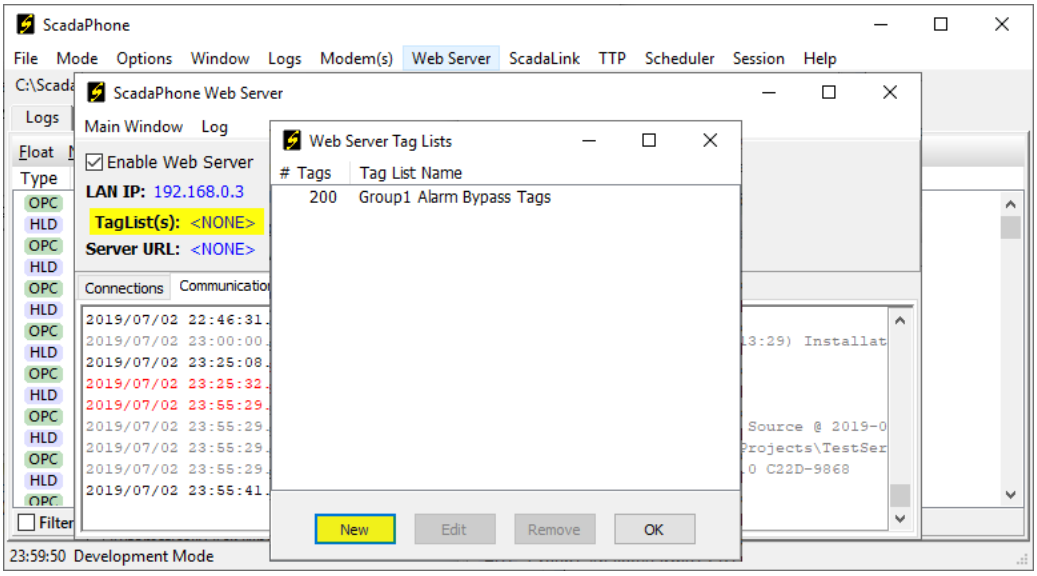
Note: Upon completion of **Step 2**, clicking on the **Discretes** tab of the ScadaPhone's main window reveals an important detail about the newly-created **RTE** tags. They are defined as **Holding Tags** in the ScadaPhone project. **Holding Tags** are *not* added to any **SCADA I/O polling lists**, but rather they are merely memory-based tags which hold a value.

This designation is indicated by the light blue icons labeled **HLD** (as opposed to SCADA-linked I/O tags which are labeled with as **OPC** in the image below):

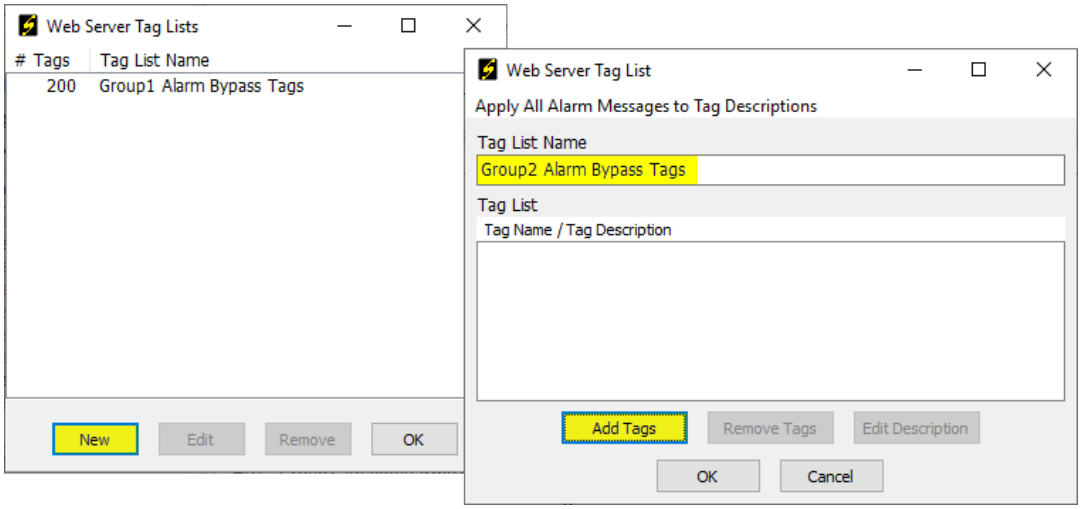


Step 3: Adding ".RTE" Tags To Web Server Tag Lists:

After the **RTE** tags have been defined to control the **Alarm Runtime Enable** fields, the next step is to make them accessible to system operators via the ScadaPhone's **Web Server Interface**. To do this, open the **Web Server** window and click on the blue hyperlink next to the **Tag List(s)** label:

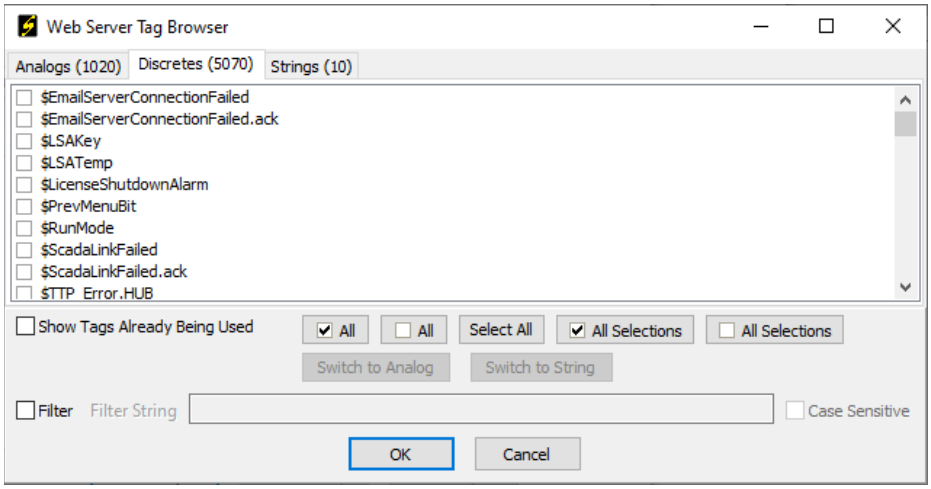


Clicking the **New** button on the **Web Server Tag Lists** window opens the **Web Server Tag List** window:



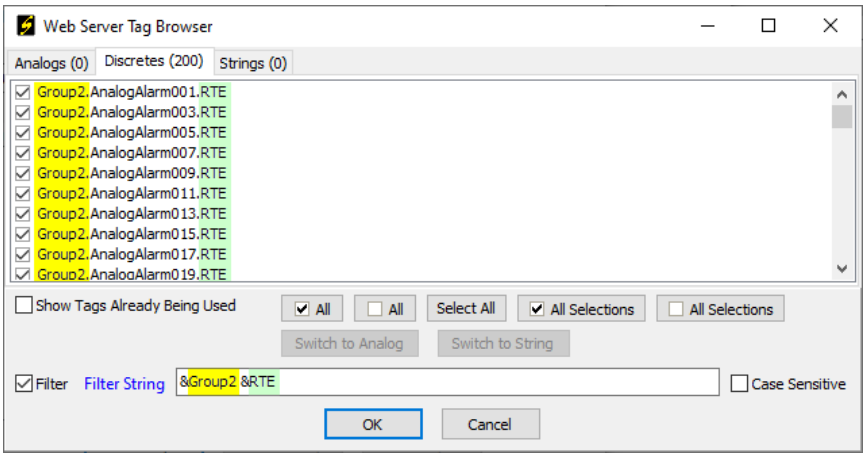
The **Web Server Tag Lists** feature supports multiple uniquely-named lists. It is important to give each list an adequately descriptive name, because that name will be presented to the operators when they connect to the **Web Server**. The best design for this **Alarm Bypass Configuration** is to segregate the **RTE Tag Lists** into one **Tag List** per **Alarm Group**. The **Tag List Name** does not have to exactly match the alarm group name, but it is best to specify a tag list name which makes it clear which alarm group is controlled by each list.

In the image above, there is an existing list for **Group1 Alarm Bypass Tags** and another list for **Group2 Alarm Bypass Tags** is being edited in the **Web Server Tag List** window. As indicated by the highlight, click the **Add Tags** button to select which tags are to be displayed for the list being edited. Clicking the **Add Tags** button opens the **Web Server Tag Browser**:

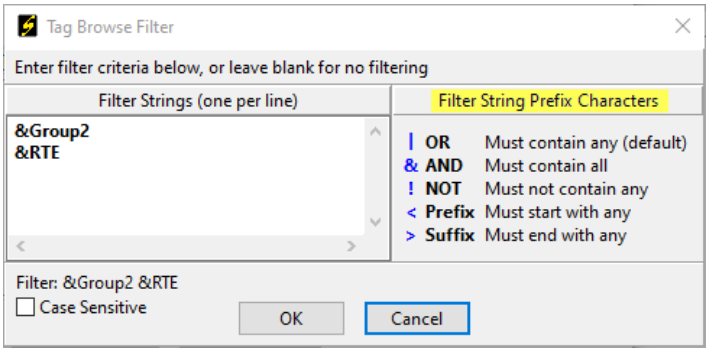


The **Web Server Tag Browser** has 3 tabs which organize the available tag names by data type: **Analog**, **Discrete** and **String**. Since all of the **RTE** tags in this example are **Discrete**, click the **Discrete** tab to view the alphabetized list of all **Discrete** tags defined in the ScadaPhone project.

In order to refine the list of tags, there is a **List Item Filter** near the bottom of the window.
By setting the **Filter String** to "**&Group2 &RTE**" the list is narrowed to the tags needed for this example.

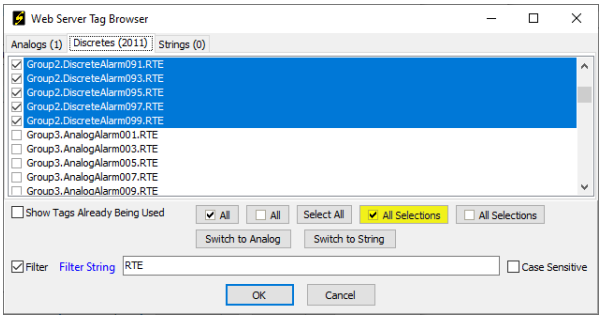


Tip: For an explanation of the **Filter** syntax, click the **Filter String** label to open the **Tag Browse Filter** window. This window assists the user in the construction of complex filters and gives an explanation of all available **Filter String Prefix Characters**.



In this example, the tag names contain the **Alarm Group** name, so the "**&Group2 &RTE"** filter narrows the list such that the **Check All** button effectively selects all desired tags; however, in most real-world projects, the tag names will not contain the **Alarm Group**, so the filtering may need to be done in phases, *or* the **Filter** can be simply reduced to "**RTE**" and the appropriate alarms can be manually selected.

When selecting tags manually, standard Windows **<Shift>** and **<Ctrl> <Left-Click>** methods can be used to select multiple items and then checked by clicking the **Check All Selections** button.

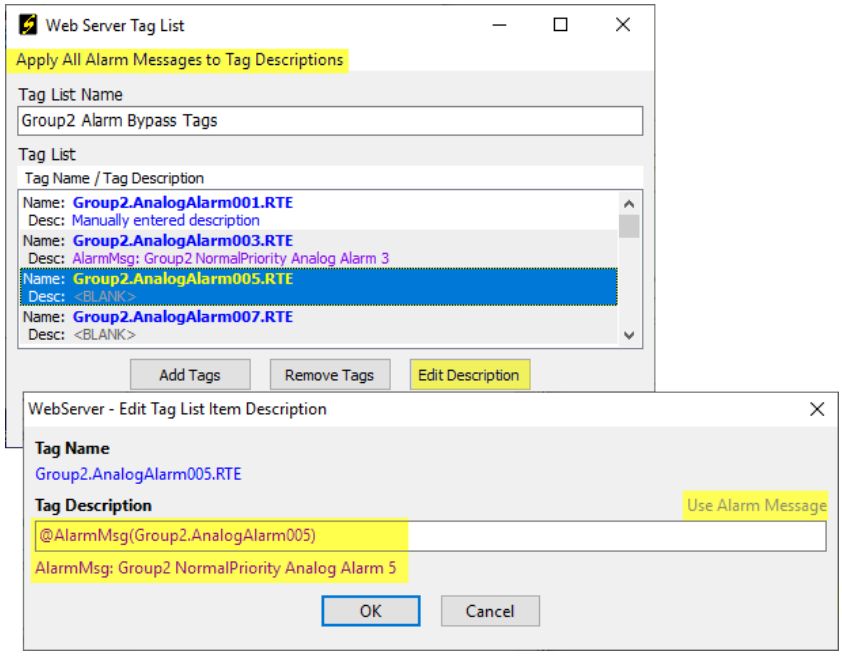


The **Web Server Tag Browser** will add all check-marked tags to the **Web Server Tag List** when **OK** is clicked.

Step 4: Adding Alarm Messages as Tag Descriptions

In many projects, the tag/alarm names are heavily abbreviated, cryptic and confusingly similar; this creates the potential for the operator to erroneously disable the wrong alarm due to misidentification. As a safeguard against this mistake, the **ScadaTEC Web Server Interface** (which is also implemented in other ScadaTEC apps) allows the system designer to enhance each **Tag List** entry with a **Description**.

Descriptive strings can be manually entered for each tag, but in ScadaPhone, there is a special enhancement to the **Tag List Item Description** feature: The associated **Alarm Message** can be used as a **Tag List Item Description**:



Note that in the **Web Server Tag List** window:

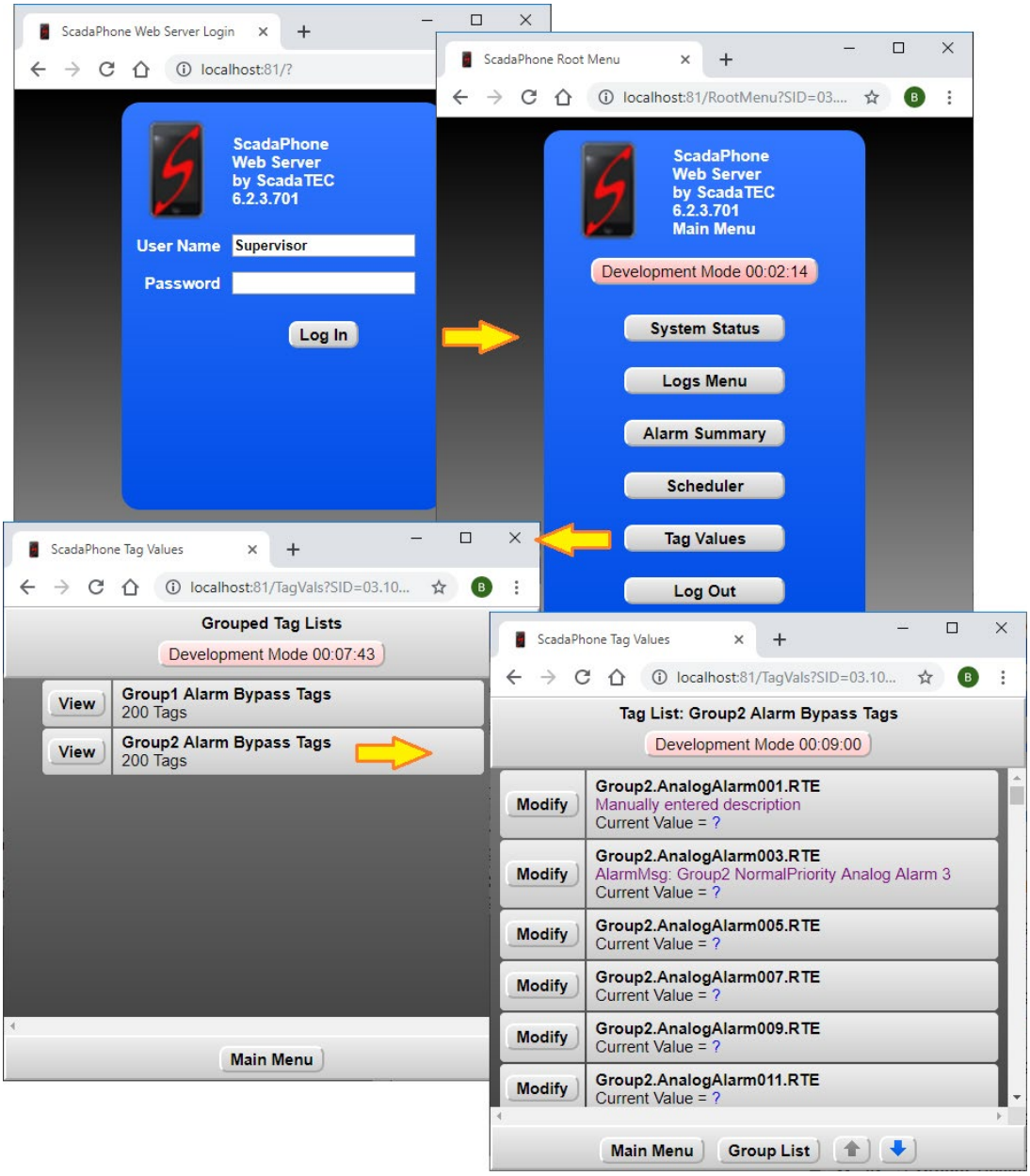
- **Manually entered** descriptions are displayed in blue.
- **Alarm Message** descriptions are displayed in purple.
- Items lacking a description are displayed with a **<BLANK>** marker.

To edit an individual description, highlight the tag name in the **Web Server Tag List** window and click the **Edit Description** button. If the highlighted tag has an associated alarm, the **Use Alarm Message** label will be displayed; clicking the **Use Alarm Message** label will insert a special **@AlarmMsg** marker into the **Tag Description** field. The **@AlarmMsg** marker requires the name of the associated alarm as a parameter. By using this special function, ScadaPhone can ensure that the description output by the Web Server will always display the current alarm message (even if the system designer changes the **Alarm Text Message** after setting up the **Web Server Tag Lists**).

Clicking the **Apply All Alarm Messages to Tag Descriptions** menu item at the top of the **Web Server Tag List** window applies **Alarm Message** descriptions to *all* **Tag List** items with one click. Do this before proceeding.

Note: After completing **Step 4**, the Web Server output of the **Tag Lists** and **Alarm Message Descriptions** can be examined by directing a local web browser (on the same computer as ScadaPhone) to load http://localhost:81 (assuming ScadaPhone's Web Server is **Enabled** and configured to listen on the default **TCP Port (81)**.

The default entry page for the **ScadaPhone Web Server** is the **Login** page. After entering a valid **ScadaPhone User** login, clicking the **Tag Values** button from the **ScadaPhone Root Menu** page opens the **Grouped Tag Lists** page. From there, click the appropriate **View** button to see the results of the **Tag List** configuration from **Step 4**:



Step 5: Configuring Direct Links to Specific Tag List Pages

In many SCADA systems, it is possible to link **HTTP URLs** to button click handlers. If the SCADA host being used in conjunction with ScadaPhone supports that capability (through **Visual Basic for Applications** or some other scripting interface), a specially formatted URL can be encoded to bypass the **ScadaPhone Web Server Login**.

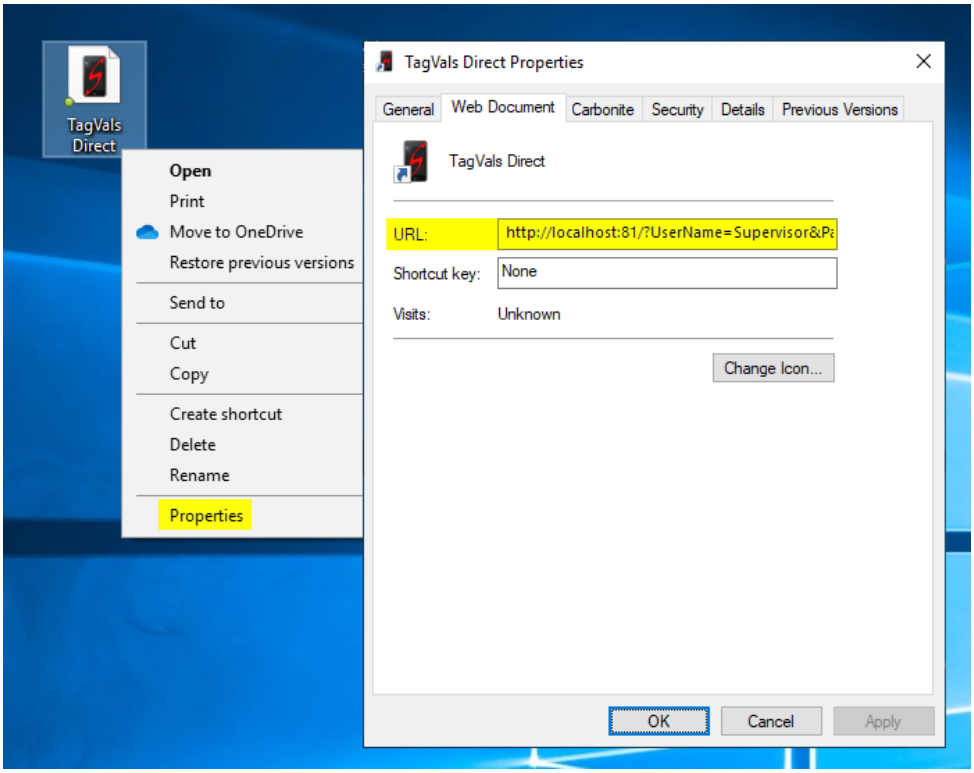
For direct access to a specific **ScadaPhone Web Server Tag List** page, format the link URL as follows:

http://**HostName:81**/?UserName=**Randy**&Password=**1234**&Page=TagVals&ExtQuery=TG=**Group2 RTE Tags**

This link assumes the following:

- ScadaPhone is running on a computer named **HostName** with the Web Server enabled on TCP Port **81**
  (Note that a raw **IP Address:Port** such as **192.168.1.10:81** can also be specified, if the port number is omitted, the **default TCP port for HTTP** [**80**] is assumed)
- The ScadaPhone project contains a **User** definition for **Randy** with login password **1234**
- There is a **Tag List** named **Group2 RTE Tags**

Links such as this can also be created on the ScadaPhone host computer by creating a **Windows Desktop Shortcut** and entering a correctly formatted link in the **URL** field:



If the system designer wants to constrain the web client from navigating to other pages once logged in to the ScadaPhone Web Server, and additional switch can be added to the end of the **Direct-Link URL**:

http://localhost/?UserName=Bob&Password=1&Page=TagVals&ExtQuery=TG=ListName%26JSV.NavEnabled=false

The switch is somewhat cryptic, but it is analyzed as follows:

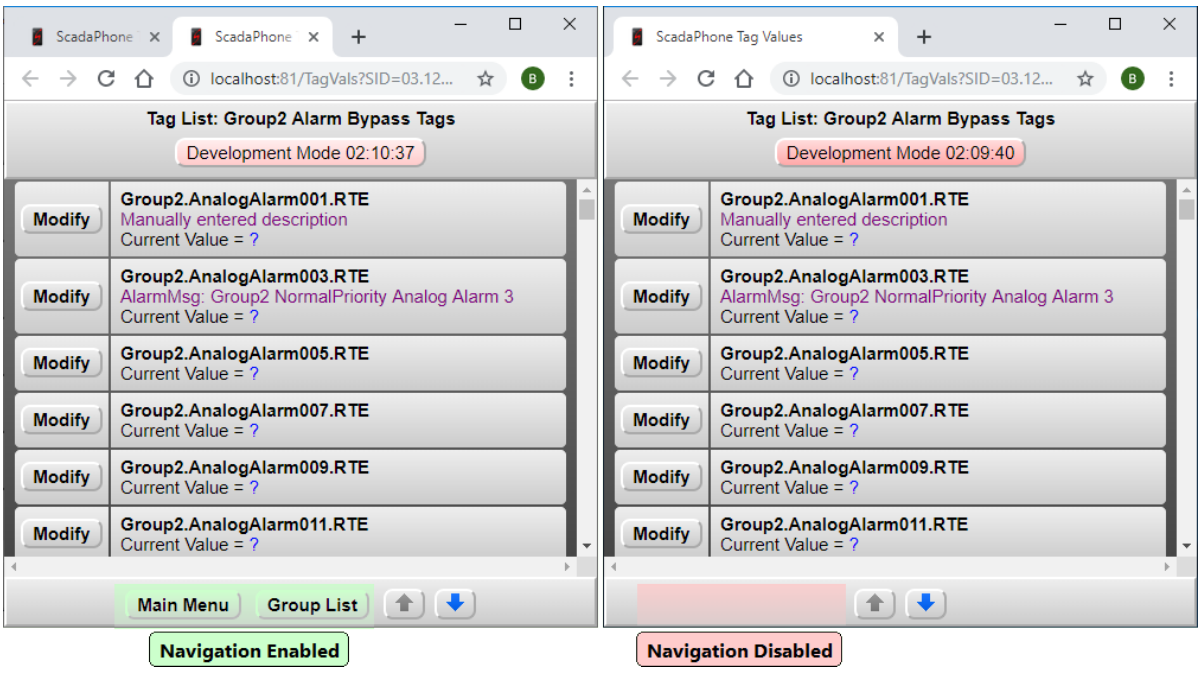
- All text after the '**?**' is referred to as the **Query String**.

- All text after the **ExtQuery=** marker constitutes the **Extended Query String**; this is treated as one Query String element by the **ScadaPhone Web Server**. If the **Extended Query String** needs to contain an ampersand (**&**), it must be encoded (or "escaped" as **%26**) to be correctly processed by the Web Server.

- The first part of the **Extended Query String** is **TG=ListName**; this identifies which **Tag List** page to display.

- The second part of the **Extended Query String** (**%26JSV.NavEnabled=false**) is treated as a second query element because it is delimited with an encoded ampersand (**%26**); after the delimiter, the assignment **JSV.NavEnabled=false** instructs the ScadaPhone Web Server to insert a **JavaScript Variable** (**JSV**) named **NavEnabled** into the HTML served to the Web Browser Client (i.e. Chrome, Firefox, Safari, etc.).

  The JavaScript in the **TagVals** page checks for a variable named **NavEnabled**, if it is defined and set to **false**, the navigation buttons at the bottom of the **Tag List** page are hidden. This prevents the operator from navigating elsewhere in the **ScadaPhone Web Server**.



The system designer can create numerous **Direct-Link URL**s which specify varying **Tag List** pages as appropriate for specific operators who are authorized to control specific alarm groups and not authorized to control other groups.

Using the **VBA** or other **scripting interface**, **Direct-Link URL**s can be constructed to relay **User Name** and **Password** credentials from the SCADA server to the **Query String** portion of the **Direct-Link URL**.

This concludes the setup procedure for the **ScadaPhone Alarm Bypass Configuration**.