Tech Note 836

Configuring an Alarm Acknowledgement Signature and Using the SignedAlarmAck() Script Function

Topic#: 002628
Created: March 2012

Introduction

SignedAlarmAck() is a script function for ArchestrA Graphics to perform an acknowledgment of one or more alarms on ArchestrA attributes that optionally require a signature depending on whether any of the indicated alarms falls within a designated priority range. If so, the user must perform an authentication of the operation to acknowledge the alarms.

Application Versions

- Wonderware InTouch® 10.5 and later
- Wonderware Application Server 3.5 and later

Before You Start

Before you complete this procedure, setup your Security in the Galaxy.

1. Create a new Galaxy
2. On the main menu, click **Galaxy-> Configure-> Security**.
3. Select **Galaxy Security**.
4. Login with the following credentials:

   • Login Name: Administrator
   • Password: (blank)

5. Repeat Step 3 and click the **Users** tab.
6. Add two users. One user is an Operator and the other is a Supervisor (Figure 1 below).

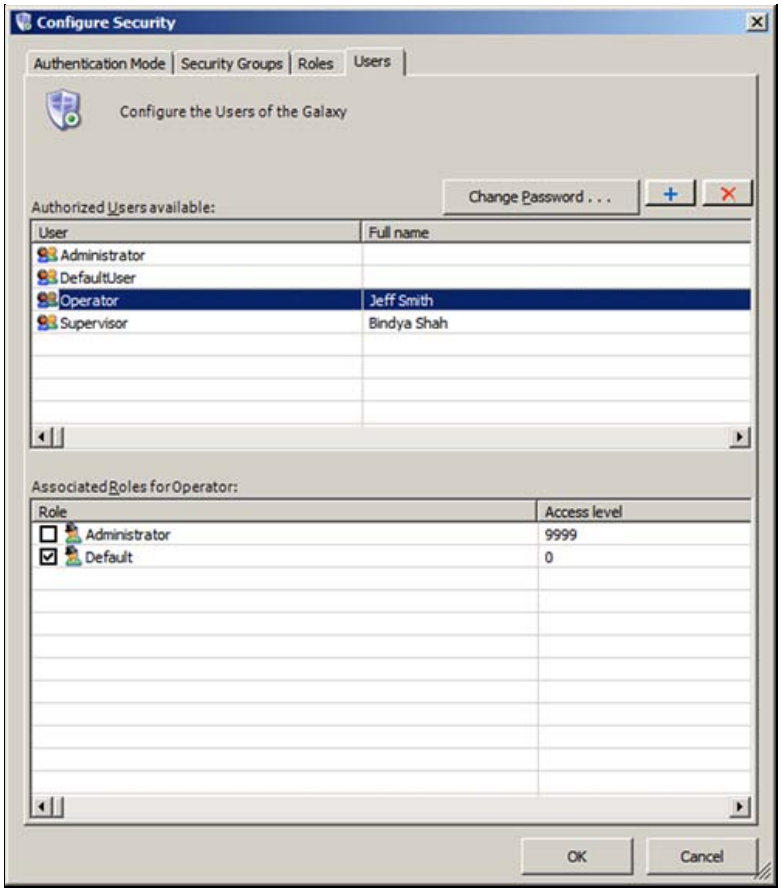

Figure 1: Add Operator and Supervisor Users

7. Highlight the Operator and click **Change Password**.
8. The old password is (blank). Set a new password, for example, **operator**.
9. Highlight the Supervisor and click the **Change Password** button.
10. The old password is (blank). Set a new password, for example, **supervior**.
11. Click the **Roles** tab.
12. Click **Default** and uncheck all options (Figure 2 below).

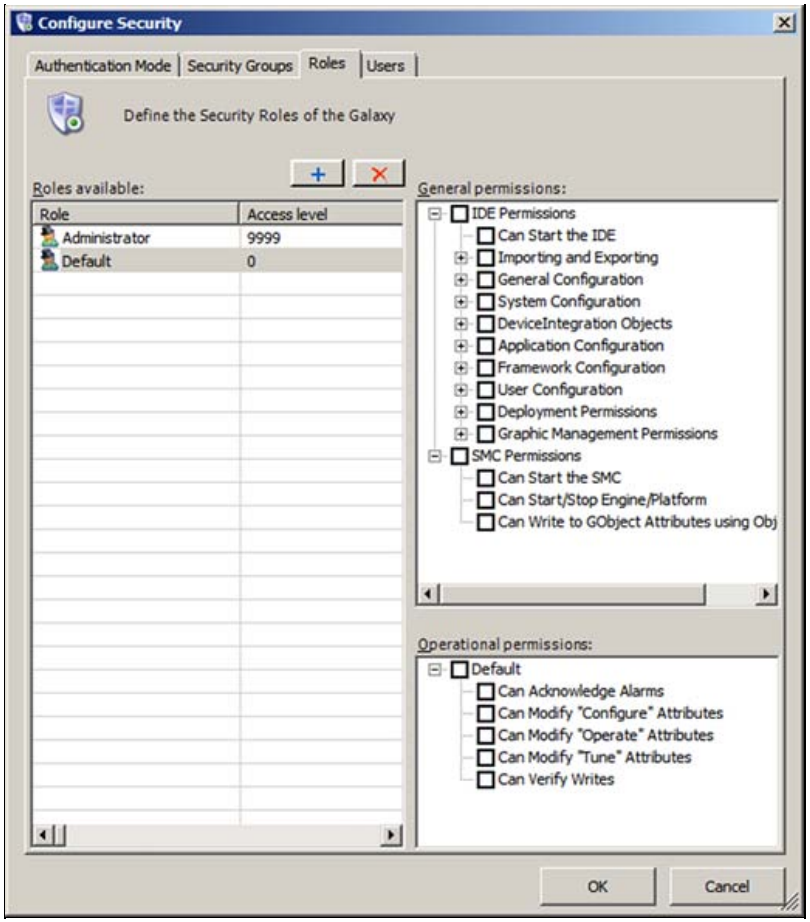

Figure 2: Uncheck all Default Security Options

13. Add the **Supervisor** and **Operator** roles with Access Levels of **9999** and **5555** respectively.
14. For the Supervisor Role, check all the permissions as shown in Figure 3 (below).

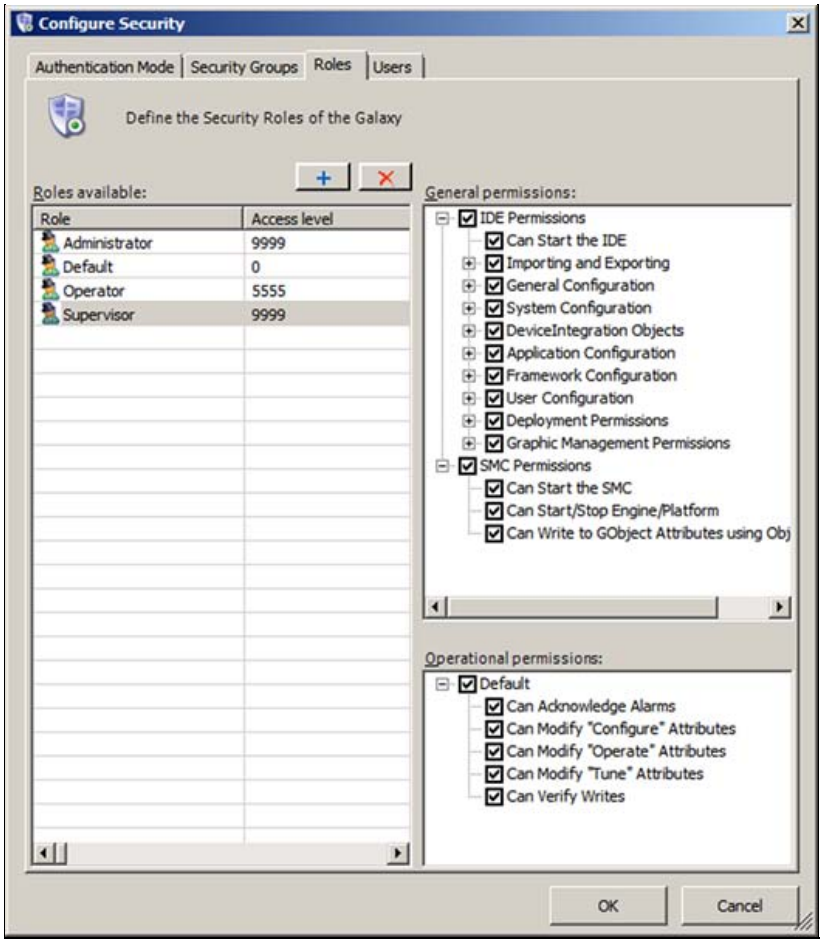

FIGURE 3: CHECK PERMISSIONS FOR SUPERVISOR

15. For the Operator role just check SMC permissions and everything under Default *except for* **Can Verify Writes** (Figure 4 below).

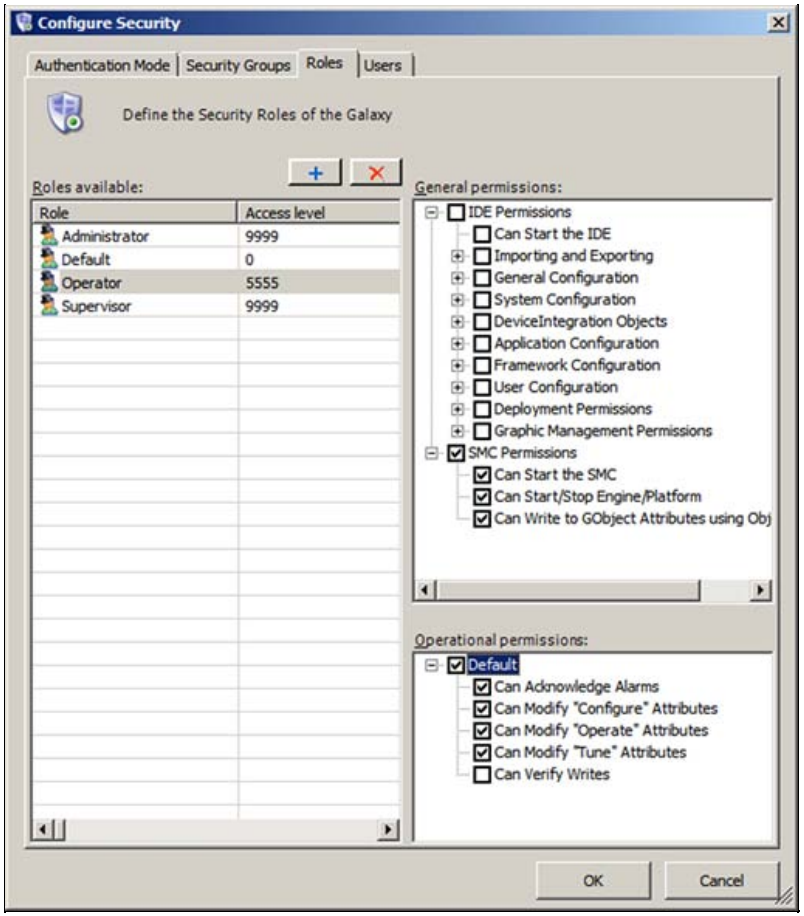

FIGURE 4: OPERATOR PERMISSIONS

16. Click the **Users** tab and make sure you have selected the associated role for each authorized user. For example, **Operator** is associated with the Operator role (Figure 5 below).

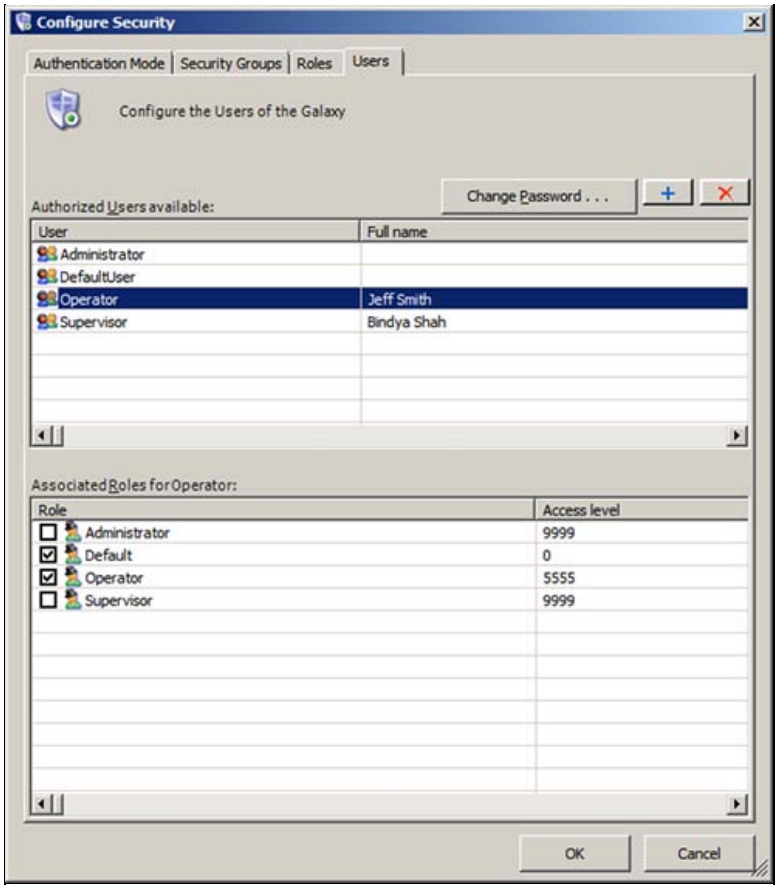

FIGURE 5: ASSOCIATED USERS AND ROLES

17. Associate the **Supervisor** with the Supervisor role (Figuer 6 below).

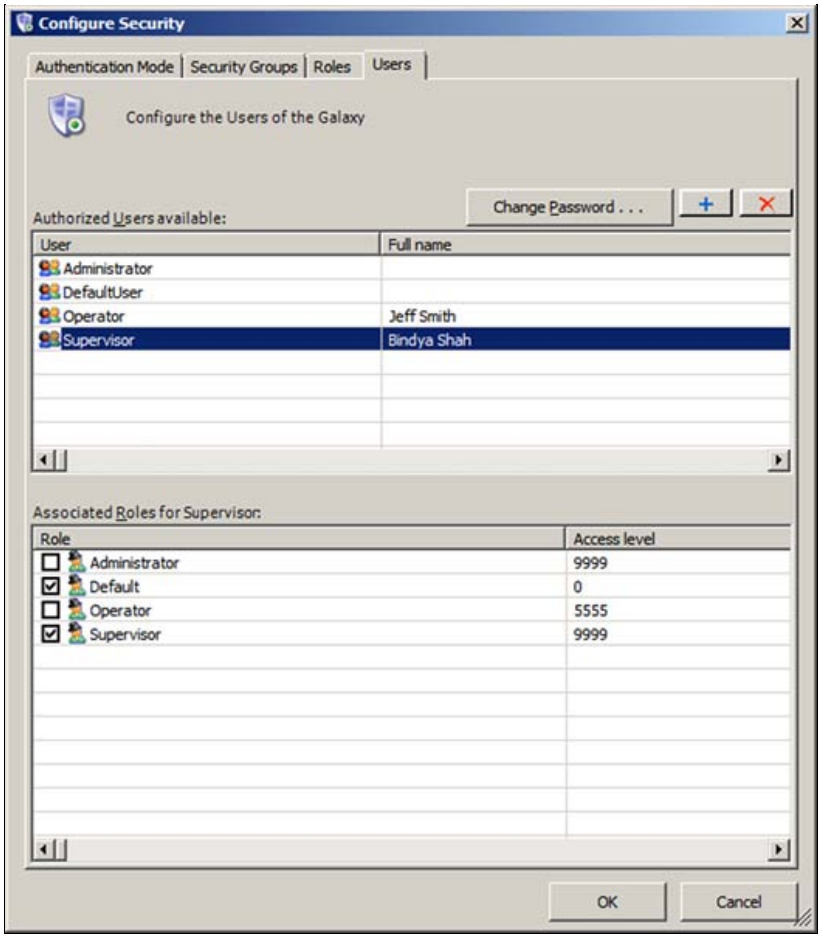

Figure 6: Associated Supervisor User and Role

Examples

This *Tech Note* includes the following examples. Each section contains script samples you can copy/paste into your Objects.

- **Configuring the signature requirement for alarm acknowledgement using the Embedded Alarm Client Control**
- **Using the SignedAlarmAck() Script Function**

Configuring the signature requirement for alarm acknowledgement using the Embedded Alarm Client Control

This example shows how to configure alarm acknowledgement signature on an Embedded Alarm Client Control.

1. Create an ArchestrA Graphic Symbol on the Graphic toolbox (e.g. AlarmSymbol).
2. Embed the Alarm Client Control on the graphic editor.
3. Double-click the Alarm Client Control and type the alarm query as **\Galaxy!Area_001**.

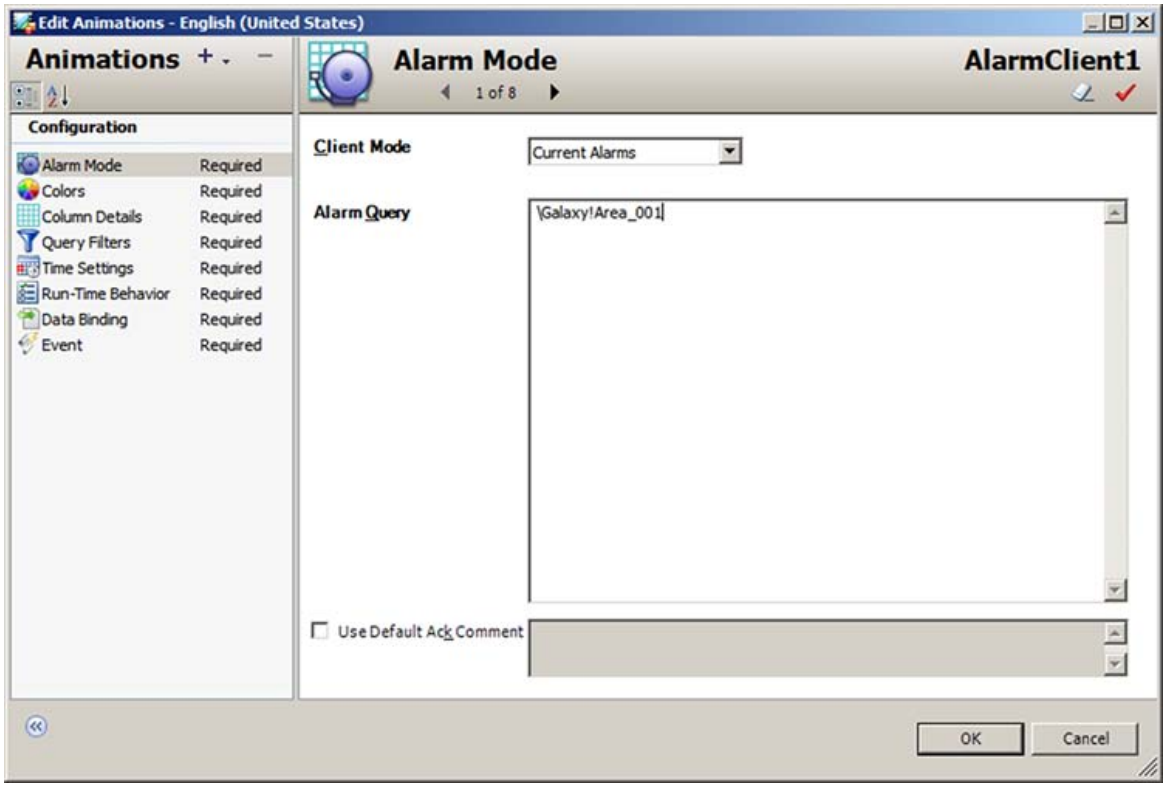

Figure 7: Alarm Client Control with Alarm Query

4. In the Run-Time Behavior configuration panel, check **Requires ACK Signature**.
5. Type the **Min Priority** and **Max Priority** for the Alarm Acknowledgement. This example uses Min Priority of **1** and Max Priority of **2.**

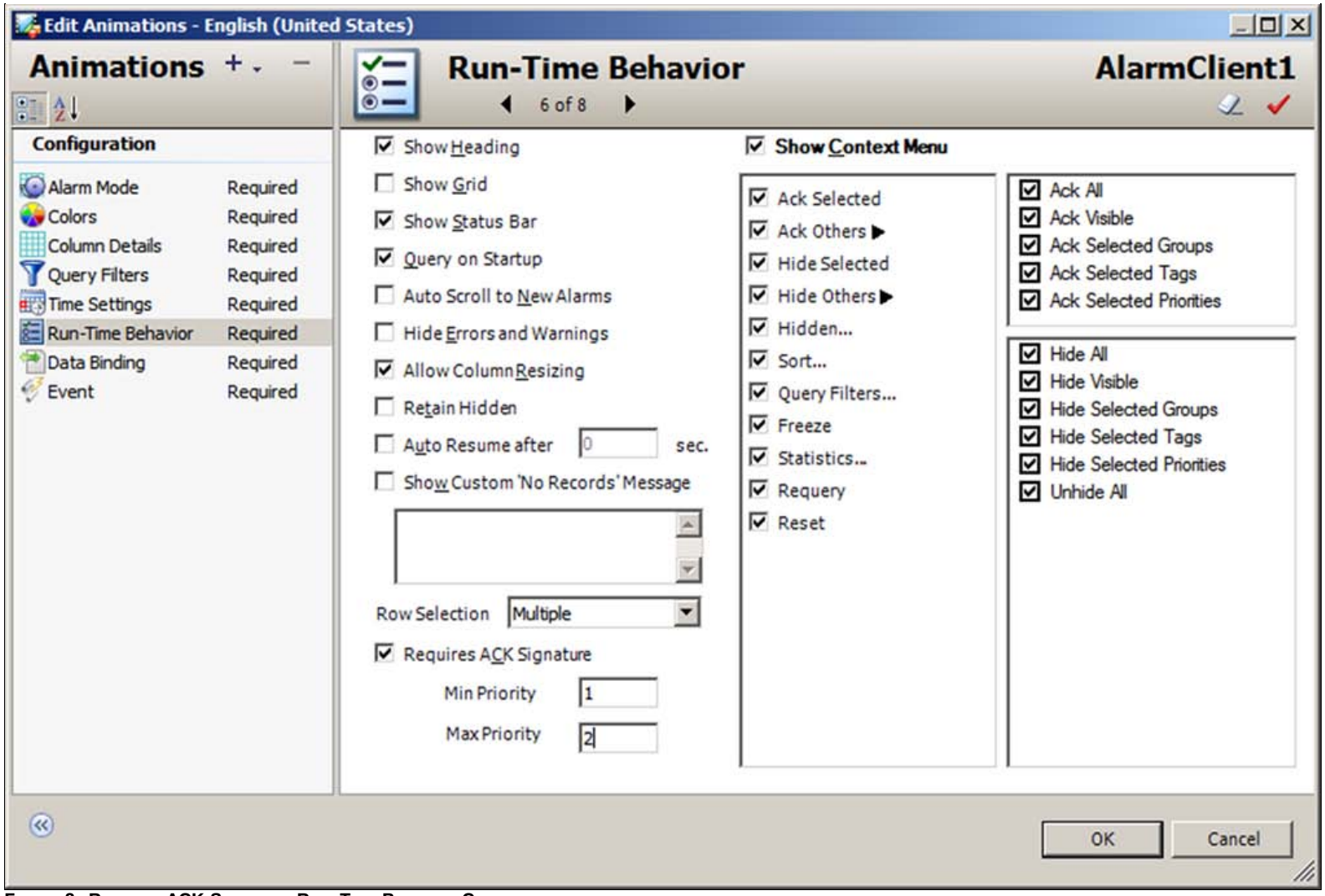

FIGURE 8: REQUIRES ACK SIGNATURE RUN-TIME BEHAVIOR CONFIGURATION

6. Open the IDE and create the following object instances:
   - $WinPlatform instance called **$WinPlatform_001**.
   - $AppEngine instance called **$AppEngine_001**.
   - $Area instance called **$Area_001**.
   - $UserDefined Object instance called **$UDO1**.
7. Enable **InTouch Alarm Provider** on the **$WinPlatform_001** object (Figure 9 below).

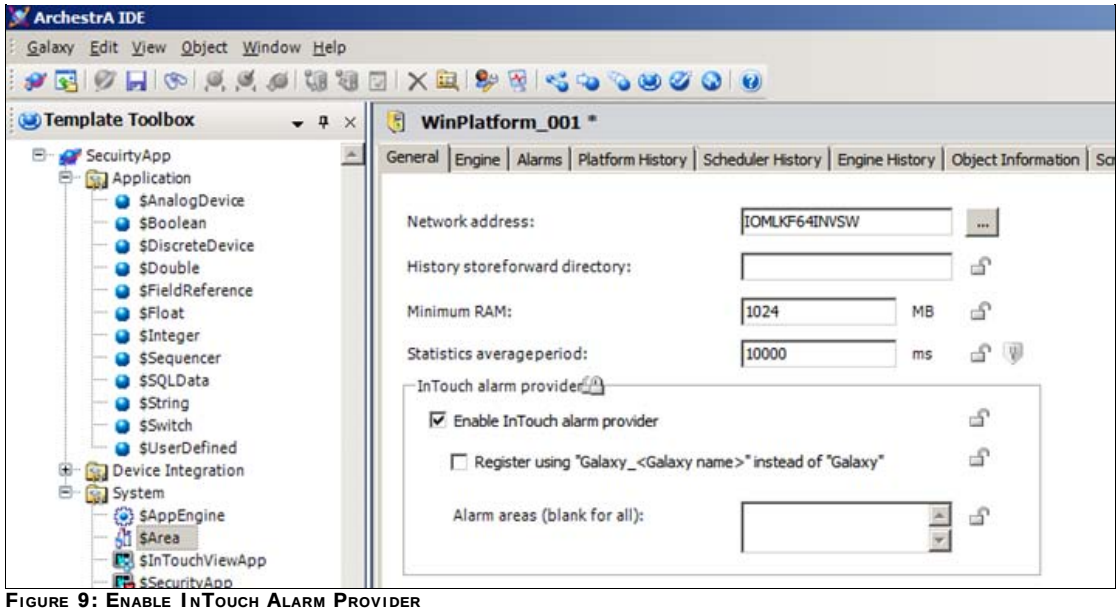

FIGURE 9: ENABLE INTOUCH ALARM PROVIDER

8. On the **$UDO1** instance, configure the following:
   - UDA called **AlarmTag** which is an **Integer** data type.
   - Analog Field Attribute called **Analog_001**.
   - Set the **Access mode** to **Input**.
   - Input Source is **UDO1.AlarmTag**.
   - Click **Enable limit alarms**.
   - For **Hi Alarm**, change the priority to **1** (Figure 10 below).

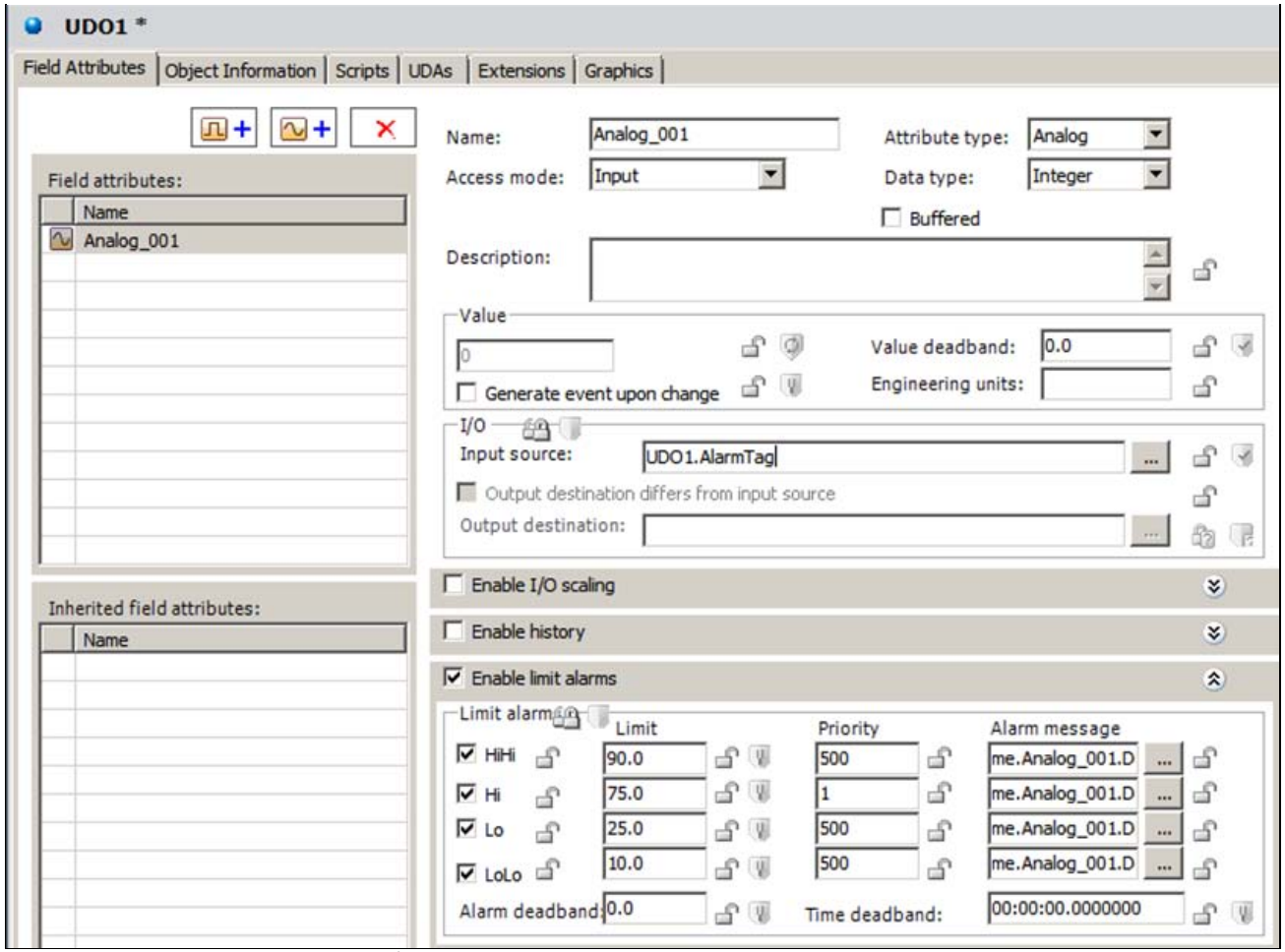

FIGURE 10: UDA CONFIGURATION FOR THE $UDO1 INSTANCE

9. Click the **Graphics** tab and create a new symbol called **EmbeddedAlarmSymbol** and then embed the AlarmSymbol that was created in step #1 above.
10. Create a new derived InTouchViewApp called **SecurityApp**.
11. Create an InTouch Window and call it **Win1**.
12. Embed the **AlarmSymbol** on the InTouch Window.
13. Embed the **SliderBasic** from the ArchestrA Symbol Library (Figure 11 below).

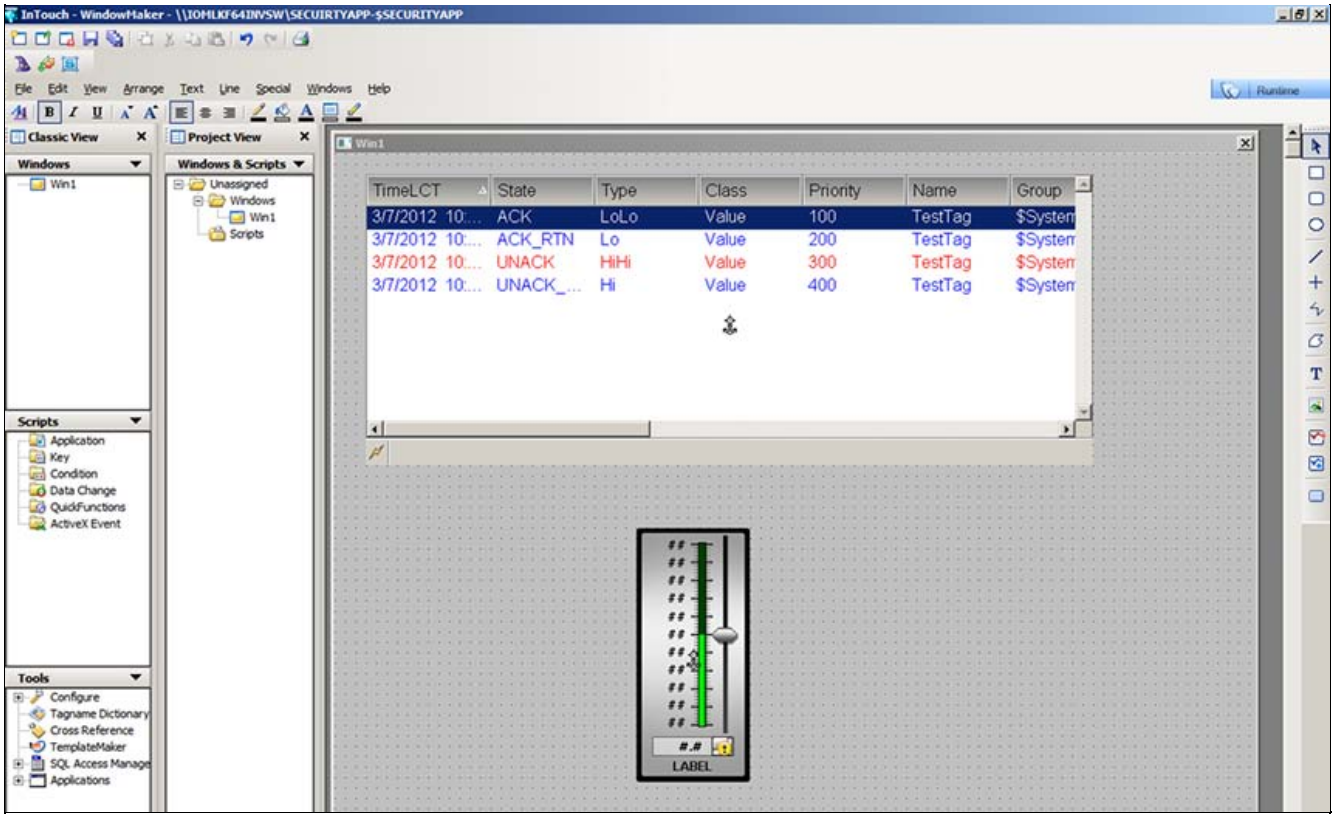

FIGURE 11: ALARMSYMBOL AND SLIDERBASIC SYMBOLS

14. Double-click the SliderBasic control and add **Galaxy:UDO1.AlarmTag** for the Custom Property's Default Value (Figure 12 blow).

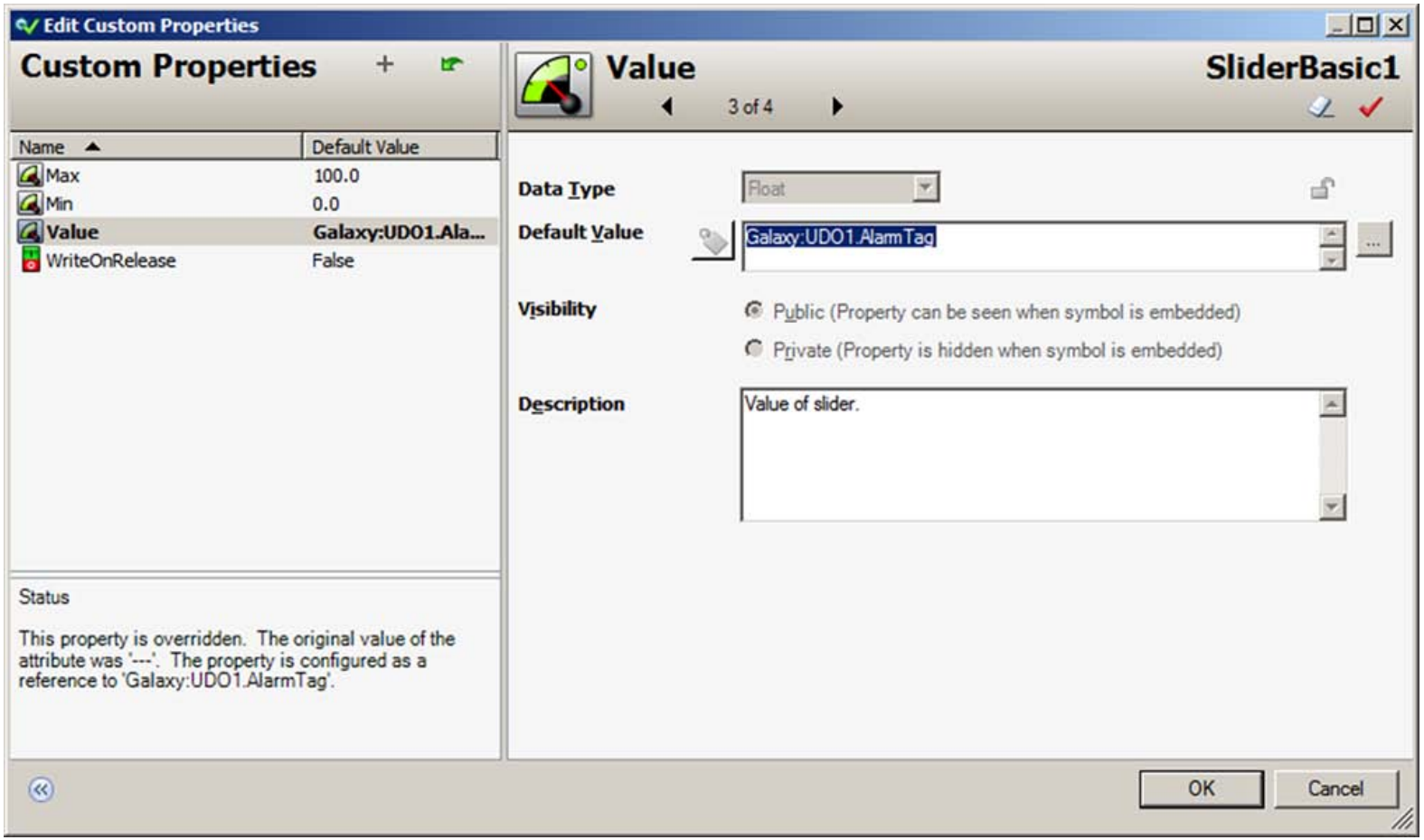

Figure 12: Custom Property Default for SliderBasic

15. Click **OK** and on the main menu, click **Special-> Security**.
16. Click **ArchestrA** for the Security Type.
17. Deploy all the objects. (WinPlatform_001, AppEngine_001, Area_001, UDO1)
18. Switch to Runtime mode.
19. Click **Special-> Security-> Log on**.
20. Login as **Supervisor** with password **supervisor**.
21. Use the Slider to reach a value where it is in Hi Alarm. This example shows a value of **85** and the Priority is **1** for that Hi Alarm.

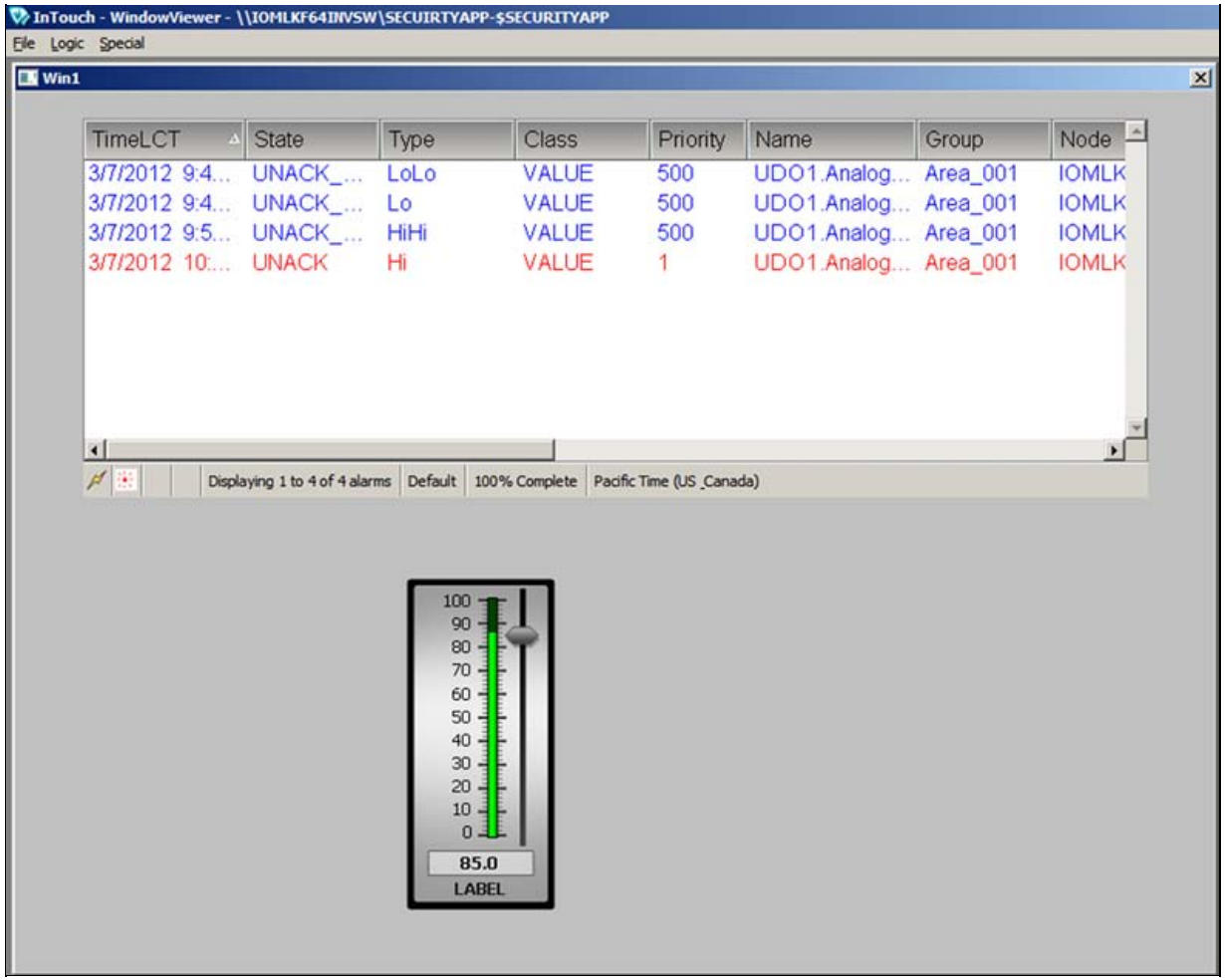

Figure 13: Slider Generating a P1 Alarm

22. Right-click on the Hi Alarm to Acknowledge it. The Ack Alarm popup appears. Type any comments and provide the username\password (Figure 14 below).

FIGURE 14: ACK ALARM REQUIREMENT

The Embedded Alarm Client control displays your comment after Acking the alarm (Figure 15 below).

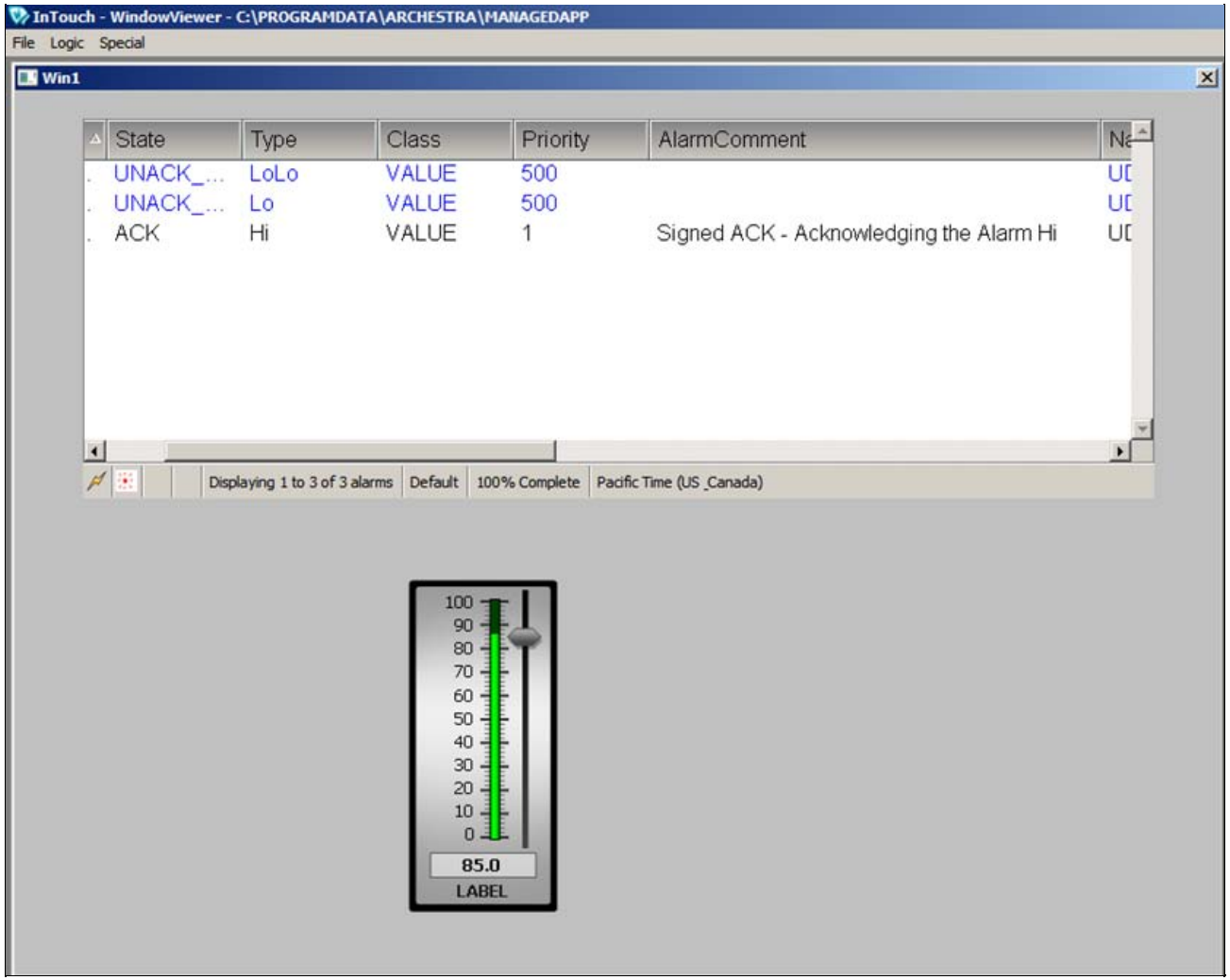

Figure 15: Comments and Ack Display

Using the SignedAlarmAck() Script Function

This example shows how to use SignedAlarmAck() script function.

1. Create an ArchestrA Graphic Symbol on the Graphic toolbox (e.g. AlarmSymbol).
2. Embed the Alarm Client Control on the graphic editor.
3. Double-click the Alarm Client Control and type the alarm query as **\Galaxy!Area_001** (Figure 16 below).

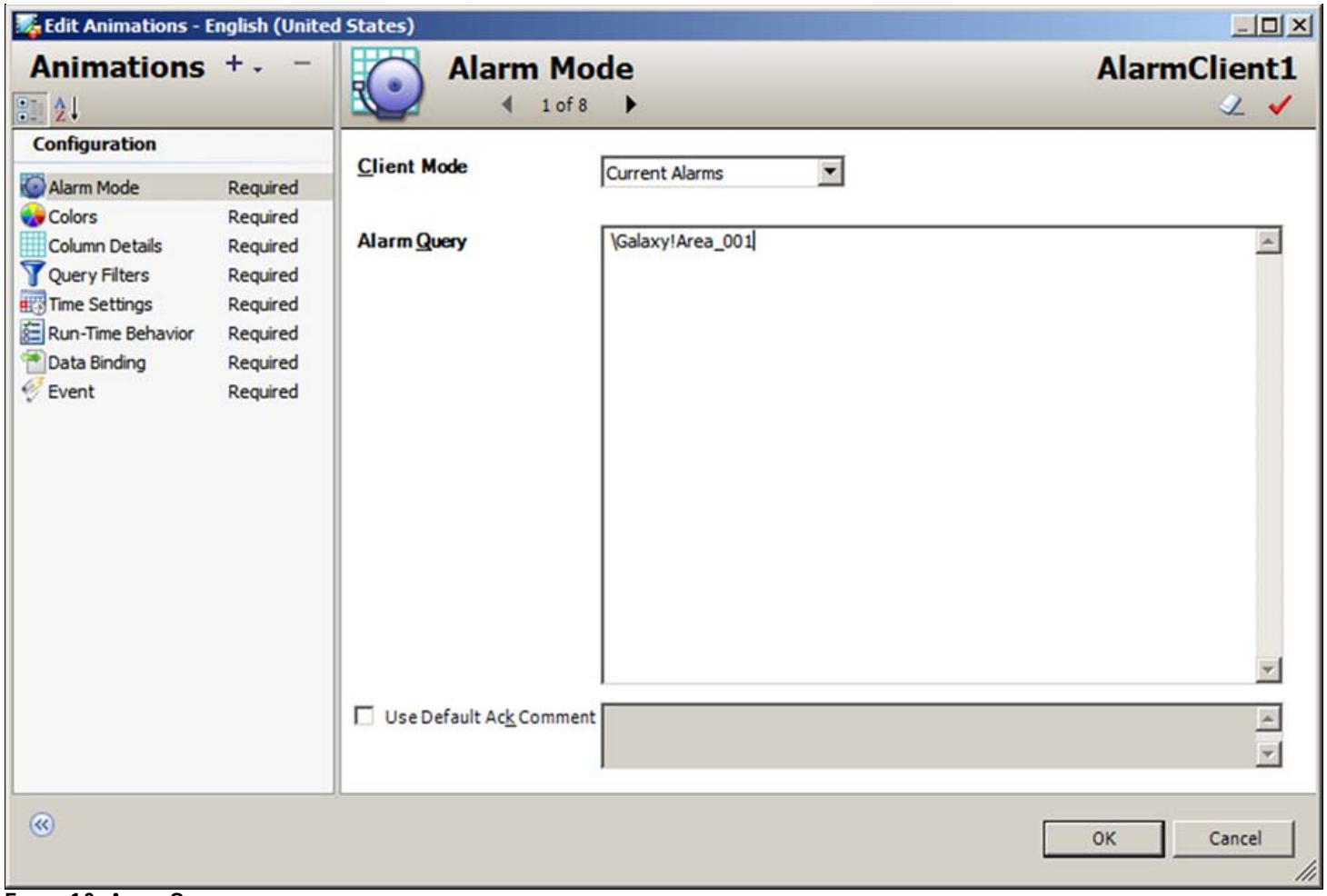

FIGURE 16: ALARM QUERY

4. Create the following object instances:
   - $WinPlatform instance called **$WinPlatform_001**
   - $AppEngine instance called **$AppEngine_001**
   - $Area instance called **$Area_001**
   - $UserDefined Object instance called **$UDO1**

5. Enable InTouch Alarm Provider on the $WinPlatform_001 (Figure 17 below).

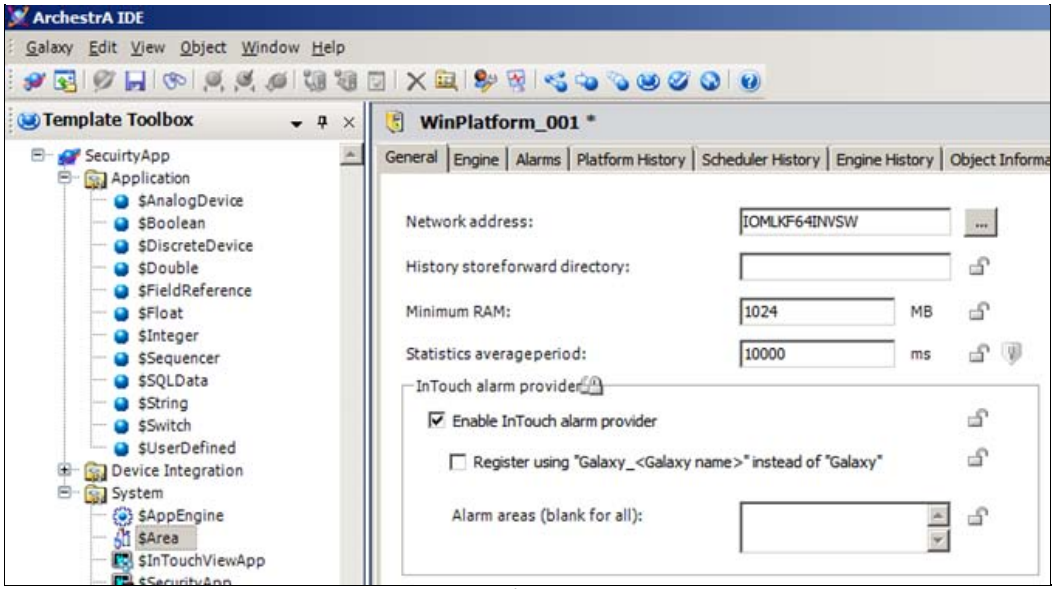

FIGURE 17: ENABLE INTOUCH ALARM PROVIDER ON THE $WINPLATFORM INSTANCE

6. On the **$UDO1** instance, configure the following:
   - UDA called **AlarmTag** which is an Integer data type.
   - Analog Field Attribute called **Analog_001**
   - Access mode is Input.
   - Input Source is **UDO1.AlarmTag**
   - Enable limit alarms
   - HiHi Alarm Alarm priority is **1**

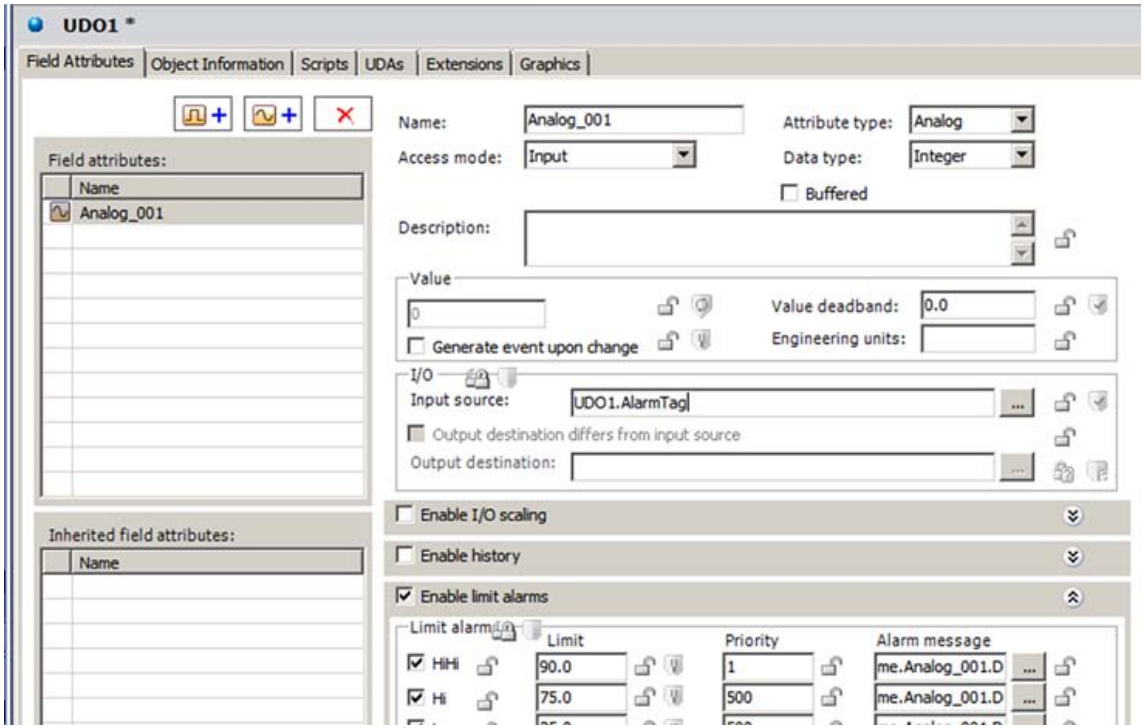

Figure 18: UDA Configuration for the $UDO1 Instance

7. Open the Alarm Symbol in Graphic Editor and add a Button called **SignedAlarmAck()**.
8. Add the following **SignedAlarmAck()** script function on an Action script animation link where the trigger is **On Left Click/Key Down**.

```
Dim Result as Integer;
Result = SignedAlarmAck("UDO1.Analog_001.HiHi", True, 1, 5, "Acked by script", False "Acknowledge Alarms by Scripting","Acknowledge HIHI Alarms");
```

This script example disables the Alarm Comment field (non-editable) provides a Title Bar caption.

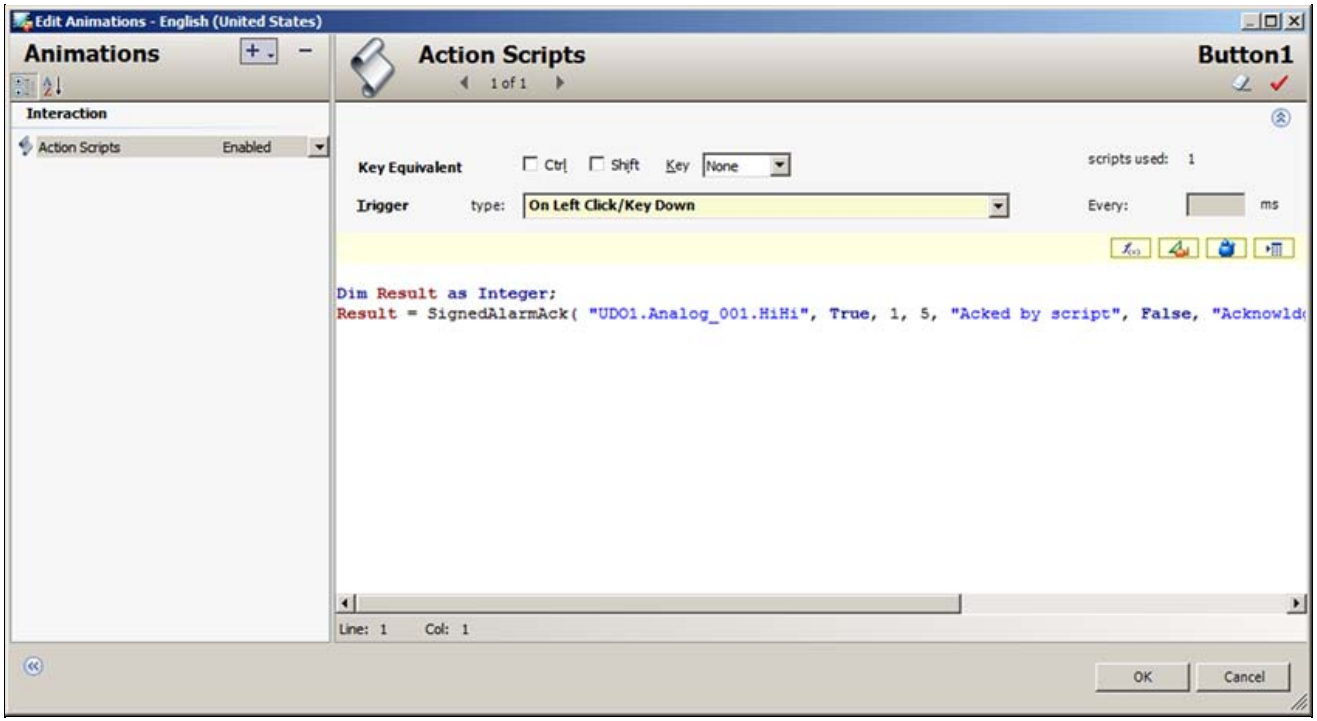

Figure 19: Action Script Example

9. Click the **Graphics** tab on the $UDO1 and create a new symbol called **EmbeddedAlarmSymbol** and embed the AlarmSymbol that was created on step # 1 above.
10. Create a new derived InTouchViewApp called **SecurityApp**.
11. Create an InTouch Window call it Win1.
12. Embed the AlarmSymbol on the InTouch Window.
13. Embed the SliderBasic from the ArchestrA Symbol Library (Figure 20 below):

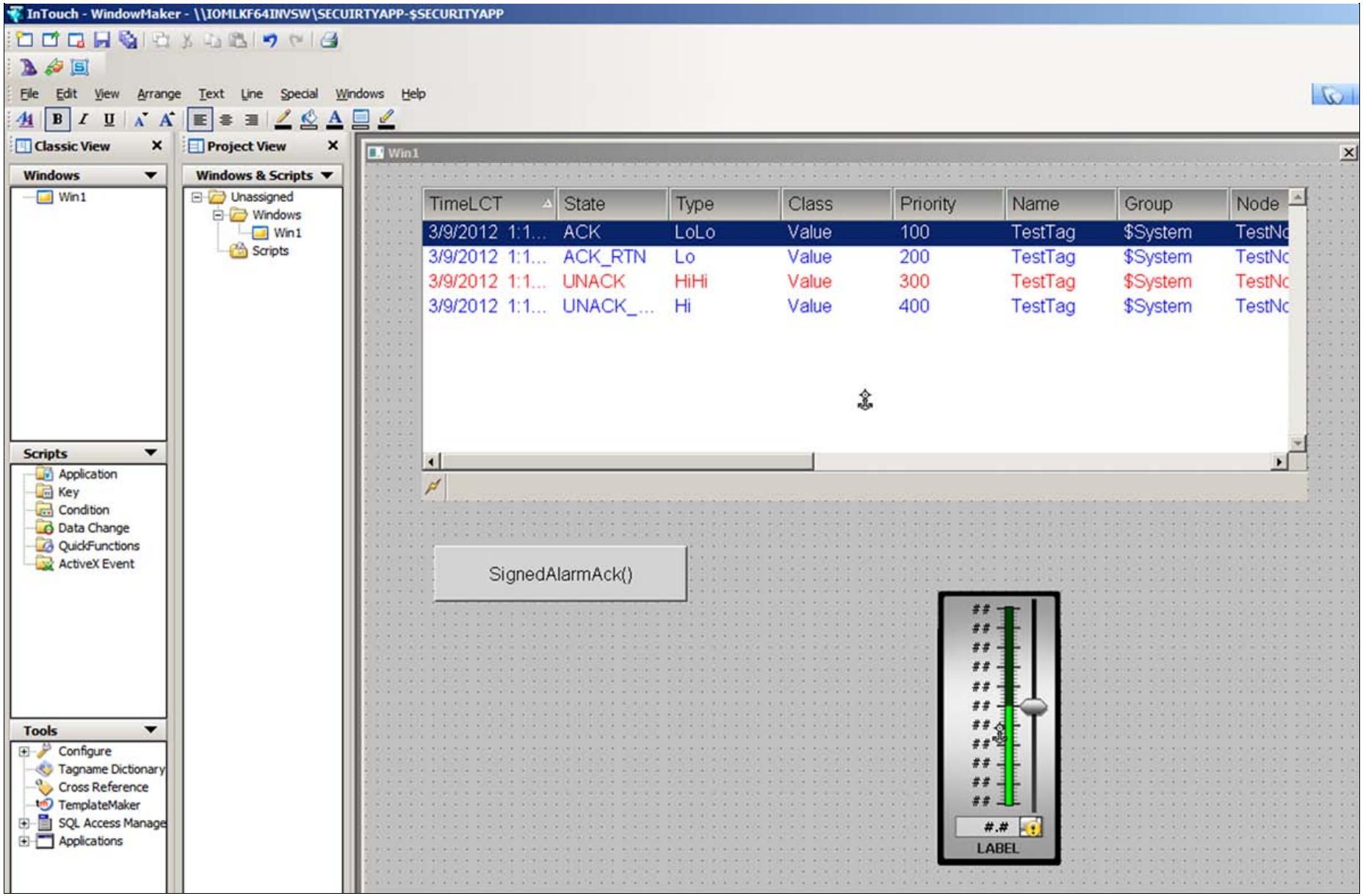

Figure 20: Slider and Button

14. Double-click the **SliderBasic** and add **Galaxy:UDO1.AlarmTag**for the Custom Property's Default Value (Figure 21 below).

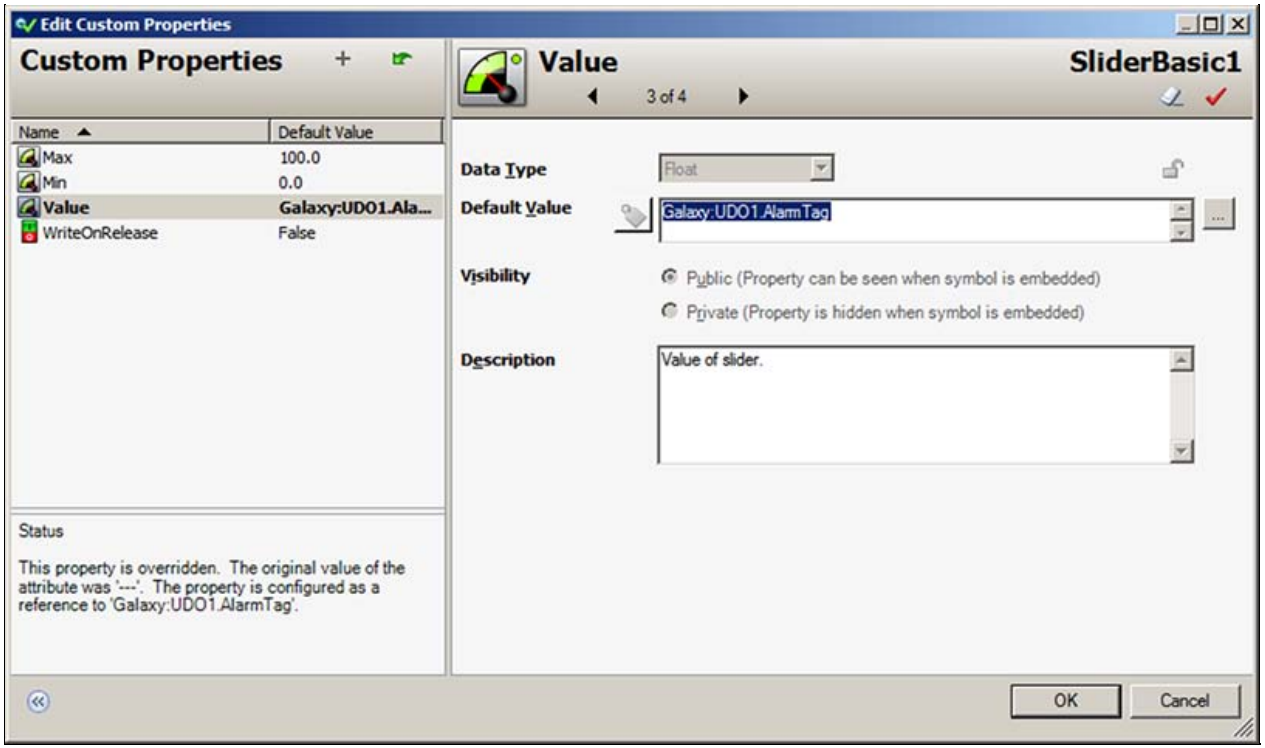

FIGURE 21: SIGNEDALARM DEFAULT VALUE

15. On the main menu, click **Special-> Security**.
16. Security Type is **ArchestrA**.
17. Deploy all the objects (WinPlatform_001, AppEngine_001, Area_001, UDO1).
18. Switch to Runtime.
19. Click **Special-> Security-> Log on**.
20. Login as **Supervisor** and provide the **supervisor** password.
21. Use the Slider to reach a value where it is in HiHi Alarm. This example shows value of **96** and the Priority is **1** for that HiHi Alarm.

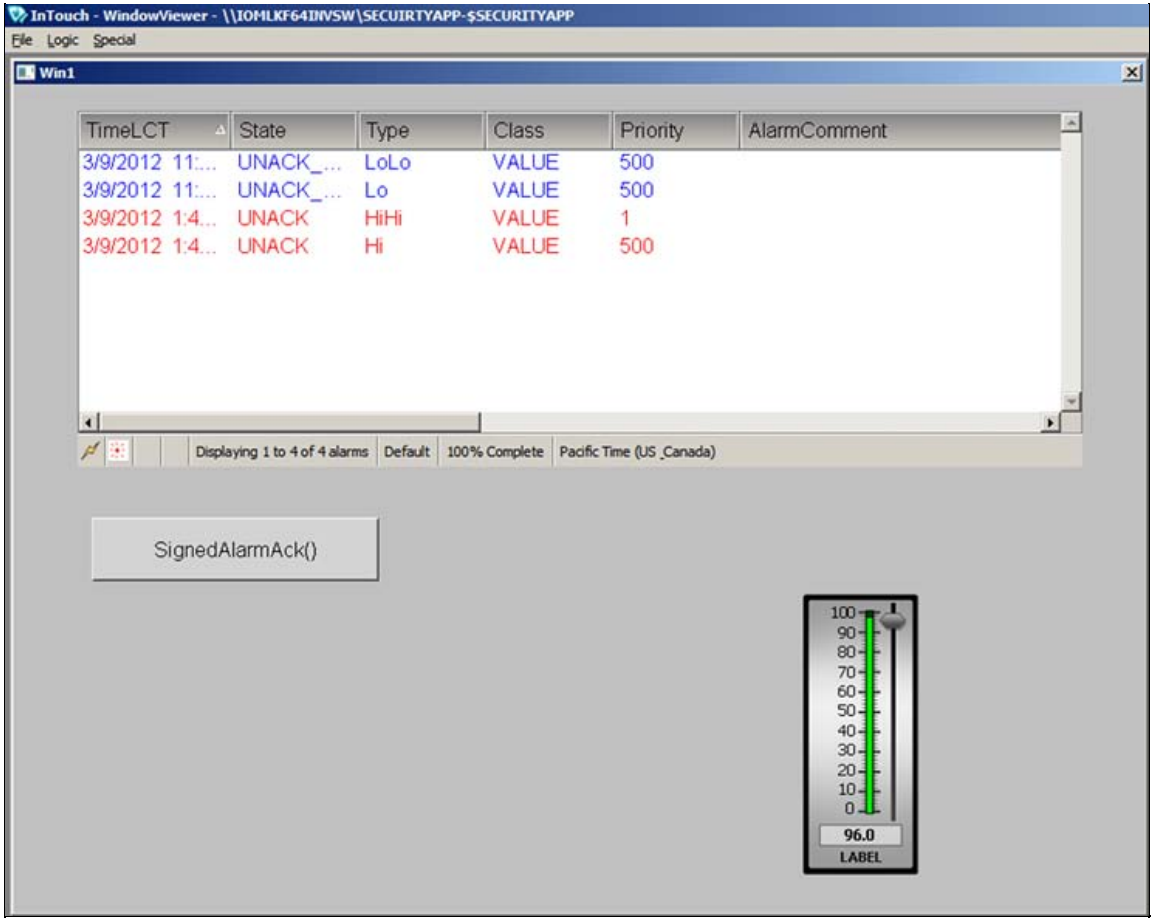

FIGURE 22: P1 ALARM USING THE SLIDER

22. Click the **SignedAlarmAck()** button. In this example, notice that you see the **Acknowldge Alarm by Scripting** dialog box where you cannot edit the alarm comment. It requires a signature to acknowledge the HiHi alarm.

Figure 23: Ack by Scripting

The comment contained in the script appears on the Embedded Alarm Client Control Display (Figure 24 below).

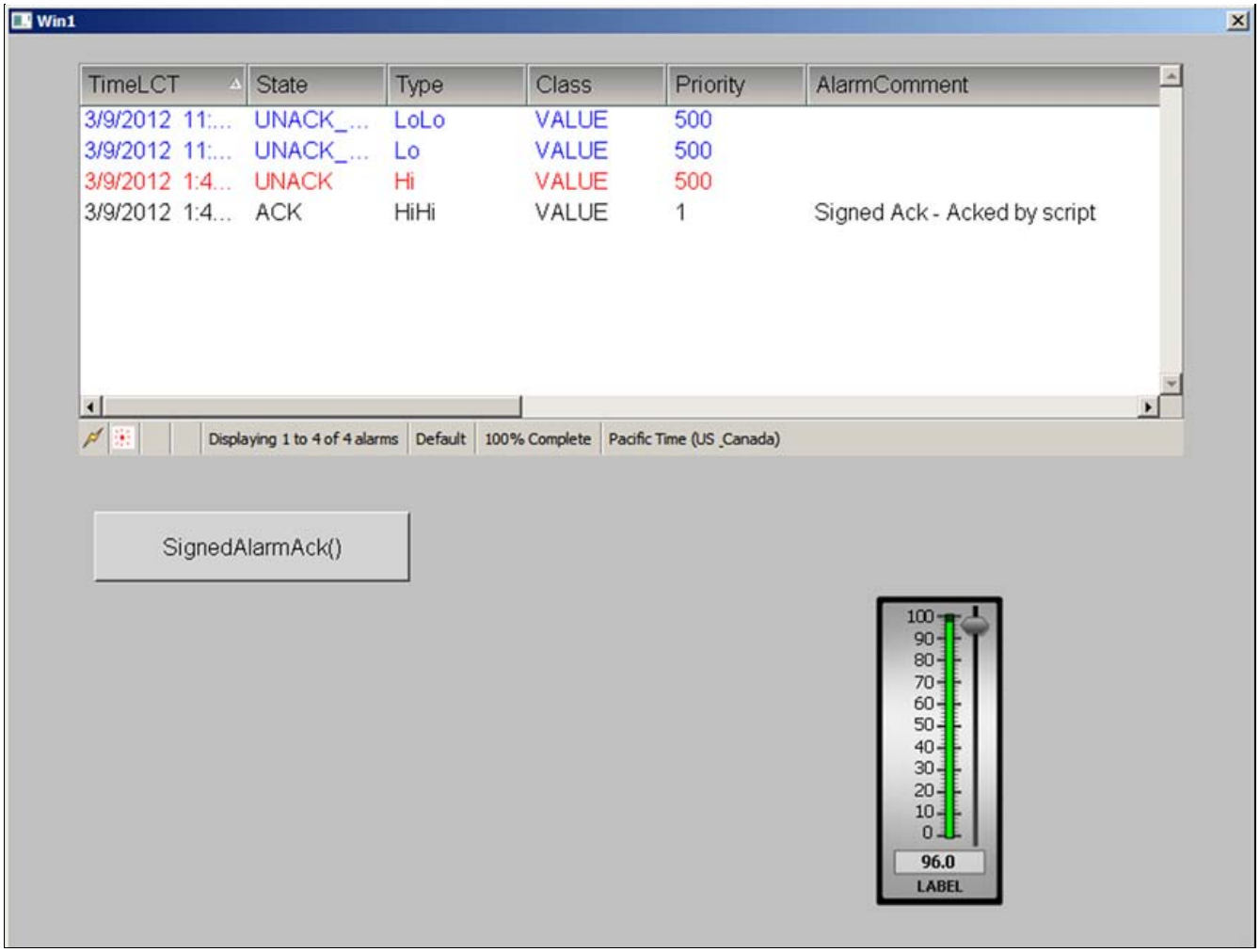

Figure 24: Acked by Script

Return values indicate success or failure status.

- A non-zero value indicates type of failure.
- For more help on the different return values, refer to the Scripting.pdf/page 57 for Wonderware Application Server 3.5.

Note: A return value of zero does not indicate if the alarms are acknowledged, only that the function wrote to the AckMsg attributes. The alarms may not be acknowledged due to insufficient permission or because the alarms have already been acknowledged.

SignedAlarmAck() Scripting Recommendations

- **SignedAlarmAck() and Alarm Configuration**

  You can use the SignedAlarmAck() function *only* in ArchestrA client scripts.

- **SignedAlarmAck() with OnShow and OnHide Scripts**

  Do not use the **SignedAlarmAck()** function with **OnShow** and **OnHide** scripts. This can cause issues with window functionality, including the window title bar, windows losing correct focus, and windows opening on top of one another.

- **SignedAlarmAck() with While True Scripts**

  Do not use the **SignedAlarmAck()** function in a **While True** script type. A signed alarm acknowledgement requires user interaction. If you want to use a While True type script, it must be set to an execution time of 30-seconds or longer to allow the user to enter the required information.

Note: The **SignedWrite()** function is supported only for client scripting and not for object scripting.

B. Shah

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