

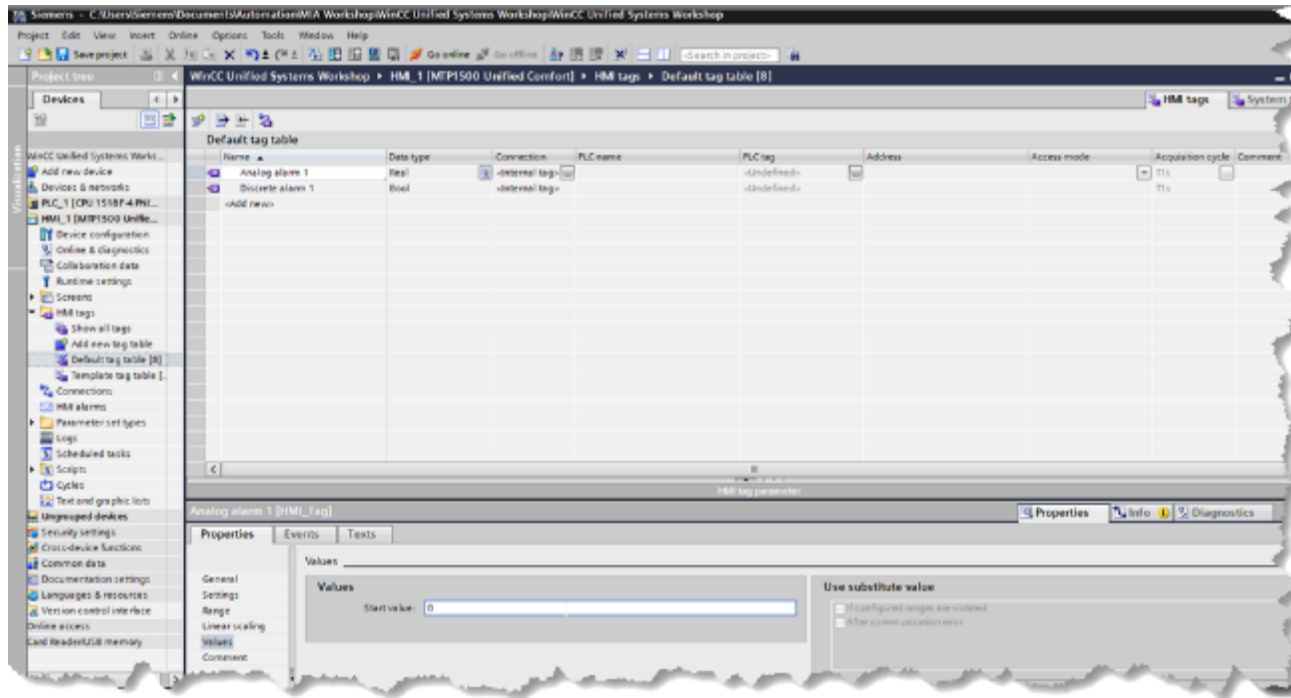
The image features a laptop in the foreground displaying the WinCC Unified Alarming software interface. The interface includes a navigation tree on the left, a central status area with a large circular gauge showing the number '30', and a right-hand panel with various data points and indicators. In the background, a semi-transparent 3D rendering of a factory production line is visible, with various machinery and conveyor belts. A prominent blue banner with white text is overlaid on the bottom left of the image. The overall aesthetic is clean and professional, using a color palette of blues, greys, and whites.

# WinCC Unified Alarming

# SIMATIC WinCC Unified

## Hands On: Analog and Discrete Alarms

# HANDS ON



1. Configure an analog and discrete alarm tag
2. Place I/O field on the screen and connect it to the variable to force them
3. Add an alarm Control to a screen
4. Download the Runtime
5. Trigger the alarm

# SIMATIC WinCC Unified Alarms

The screenshot shows the SIMATIC WinCC Unified Alarms configuration interface. The Project Tree on the left shows the hierarchy: Workshop\_Alarm2 > HMI\_1 [MTP700 Unified Basic] > HMI alarms. The main window displays the 'Discrete alarms' configuration table. A table with 10 columns (ID, Name, Alarm text, Alarm class, Trigger tag, Triggers, Connection of t..., Acknowledg..., Ackn..., Acknowledg..., Connection stat...) shows one alarm with ID 1, Name 'Discrete alarm\_1', and Alarm text 'Alarm Single Triggered'. The Trigger tag is set to '<No tag>'. A 'Properties' dialog box for 'HMI\_Tag\_1 [HMI\_Tag]' is open, showing the 'General' tab with fields for Name (AlarmSingle), PLC tag (<Undefined>), Connection (<Internal tag>), PLC name, Address, and Access mode. The 'Settings' tab shows Data type: Bool, Length: 1, and HMI data type: Bool. A 'New HMI Tags' dialog box is also open, showing a table with columns Name, Data type, and Address. The table contains two entries: 'RunPB\_vlv1' and 'RunPB\_vlv2', both with Data type 'Bool'. The interface includes various tabs like 'Discrete alarms', 'Analog alarms', 'OPC UA A&C', 'System events', and 'Alarm classes'. The Project Tree also shows 'Screens', 'Screen management', 'HMI tags', 'Connections', and 'Parameter set types'.

ID	Name	Alarm text	Alarm class	Trigger tag	Triggers	Connection of t...	Acknowledg...	Ackn...	Acknowledg...	Connection stat..
1	Discrete alarm_1	Alarm Single Triggered	Alarm	<No tag>	0		<No tag>	0		

Name	Data type	Address
None		
RunPB_vlv1	Bool	
RunPB_vlv2	Bool	

1. Open HMI alarms from Project Tree
2. Add a new Discrete alarm
3. Assign a Trigger Tag
4. Create a new Tag, place it in the 'New HMI Tags' folder
5. Name tag as shown – use Data type 'Bool'
6. Add alarm text as shown

# SIMATIC WinCC Unified Alarms

The screenshot shows the SIMATIC WinCC Unified Alarms configuration interface. The main window displays a table of discrete alarms with columns for ID, Name, Alarm text, Alarm class, Trigger tag, Trigger bit, Connection of tag, Acknowledgment, and Connection status. Two alarms are listed: 'Discrete alarm\_1' and 'Discrete alarm\_2'. The 'Discrete alarm\_2' row is highlighted, and its 'Trigger tag' is set to 'AlarmByte' and 'Trigge..' is set to '1'. A callout '1' points to the '+ Add new' button. A callout '2' points to the 'Discrete alarms' tab. A callout '3' points to the 'New HMI Tags' folder in the tree view. A callout '4' points to the 'AlarmByte' tag configuration dialog, showing the name 'AlarmByte', data type 'Byte', length '1', and HMI data type 'Bool'. A callout '5' points to the 'AlarmByte' tag in the table.

ID	Name	Alarm text	Alarm class	Trigger tag	Trigge..	Connection of t..	Acknowledg...	Ackn...	Acknowledg...	Connection stat..
1	Discrete alarm_1	Alarm Single Triggered	Alarm	AlarmSingle	0		<No tag>	0		
2	Discrete alarm_2	Alarm Byte - Bit 1 Triggered	Alarm	AlarmByte	1		<No tag>	0		
<Add new>										

AlarmByte [HMI\_Tag]

General

Name: AlarmByte

Data type: Byte

Length: 1

HMI data type: Bool

PLC tag: <Undefined>

Connection: <Internal tag>

PLC name:

Address:

Access mode:

Settings

Name

Data type

Address

None

AlarmByte

AlarmSingle

RunPB\_vlv1

RunPB\_vlv2

Byte

Bool

Bool

Bool

Show all

Edit

Create

1. Add a new Discrete alarm
2. Assign a Trigger Tag
3. Create a new Tag, place it in the 'New HMI Tags' folder
4. Name tag as shown – use Data type 'Byte'
5. Add alarm text as shown – Change Trigger Bit = 1

# SIMATIC WinCC Unified Alarms

The screenshot shows the SIMATIC WinCC Unified Alarms configuration interface. The main window displays the 'Analog alarms' tab, which is highlighted with a green circle labeled '1'. Below the tab, a table lists the configured alarms. The first alarm, 'Analog alarm\_1', is highlighted with a green circle labeled '2'. The 'Trigger tag' column for this alarm is set to '<No tag>', which is highlighted with a green circle labeled '3'. The 'Limit' column is set to '75', and the 'Limit mode' is set to 'Higher', both highlighted with green circles labeled '6'. A 'Properties' dialog box is open for the 'Analog alarm\_1' tag, showing the 'General' tab. The 'Name' field is 'AlarmAnalog', the 'PLC tag' is '<Undefined>', and the 'Connection' is '<Internal tag>'. The 'Settings' section shows 'Data type' as 'Int', 'Length' as '2', and 'HMI data type' as 'Int', with a green circle labeled '5' next to the 'Data type' field. A 'New HMI Tags' dialog box is also open, showing a list of tags under the 'New HMI Tags [9]' folder. The 'AlarmByte' tag is selected, with a data type of 'Byte' and an address of 'None', highlighted with a green circle labeled '4'. The 'Edit' and 'Create' buttons are visible at the bottom of the dialog.

ID	Name	Alarm text	Alarm class	Trigger tag	Connection of t..	Limit	Limit mode
3	Analog alarm_1	Analog Threshold Reached	Alarm	<No tag>		75	Higher
<Add new>							

Name	Data type	Address
None		
AlarmByte	Byte	
Status_vlv1	Int	
Status_vlv2	Int	

1. Select Analog alarms tab
2. Add a new Analog alarm
3. Assign a Trigger tag
4. Create a new tag in the 'New HMI Tags' folder
5. Add tag as shown
6. Add alarm text as shown, change limit as shown

# SIMATIC WinCC Unified Alarms

1. Open Alarm screen
2. Drag Alarm view from Controls Toolbox as place on screen as shown
3. Change Appearance as shown

The screenshot displays the SIMATIC WinCC environment. On the left, the Project tree shows the 'Workshop\_Alarm2' project with 'HMI\_1 [MTP700 Unified Basic]' selected. The 'Screens' folder is expanded, and the 'Alarm' screen is highlighted. A green circle with the number '1' is placed over this selection. The central workspace shows the 'Alarm' screen layout, which includes a table with columns for 'ID', 'Raise time', and 'Alarm text'. Below the table is a toolbar with various icons. A green circle with the number '2' is placed over the 'Alarm' icon in the Controls toolbox on the right. The Properties window at the bottom shows the 'Appearance' section, where the 'Appearance - style item' is set to 'HmiAlarmControl'. A green circle with the number '3' is placed over this dropdown menu.

ID	Raise time	Alarm text
1		
2		
3		
4		
5		

Name	Static value	Dynamization (0)
General		
Appearance		
Acknowledgment alarms ...	Medium	None
Appearance - style item	HmiAlarmControl	
Background - color	204, 204, 204	None



# SIMATIC WinCC Unified Alarms

1. Drag Text and IO fields as shown for each tag as shown
2. Highlight New HMI Tags folder and drag each tag from the details view to each IO field

**Note: By dragging the tags, they are automatically filled in for the Process Value**

The screenshot displays the SIMATIC WinCC Unified Alarms software interface. On the left, the Project tree shows the hierarchy for 'Workshop\_Alarm2', with 'New HMI Tags [9]' highlighted. The main workspace shows an alarm table with 5 rows and columns for ID, Raise time, and Alarm text. Below the table are HMI design elements: 'Alarm Single', 'Alarm Analog', and 'Alarm Byte'. The 'IO field\_1 [IO field]' properties panel is open, showing the 'Process value' dropdown set to 'Tag'. The 'Tag' properties panel shows 'AlarmSingle' selected for the 'Tag' field. The 'Details view' at the bottom left shows a table of tags:

Name	Data type	Address
AlarmAnalog	Int	
AlarmByte	Byte	
AlarmSingle	Bool	
Name vlv1	WString	

Red circles with numbers '1' and '2' are overlaid on the image. Circle '1' points to the 'Alarm Single' and 'Alarm Analog' fields in the design area. Circle '2' points to the 'New HMI Tags [9]' folder in the Project tree and the 'AlarmSingle' tag in the Details view.

# SIMATIC WinCC Unified Alarms

The screenshot shows the SIMATIC WinCC Unified Alarms interface. The Project Tree on the left (2) shows the HMI device selected. The main window (3) displays the Alarm list with two entries: 'Alarm Byte - Bit 1 Triggered' and 'Analog Threshold Reached'. The Acknowledgment buttons (4) are visible at the bottom. The browser window (6) is open in the background.

ID	Raise time	Alarm text
	2/8/2024 8:18	Alarm Byte - Bit 1 Triggered
	2/8/2024 8:19	Analog Threshold Reached

Alarm Analog: 76

Alarm Byte: 0000 0010

1. Save the project
2. Select HMI device in the Project Tree
3. Press Start Simulation, Portal will open the default browser
4. Test your work
5. Use the built-in Acknowledgment buttons
6. When finished close the browser

**Note: For the Byte Alarm, we selected Bit 1. So we enter binary '10' to trigger the alarm**



# SIMATIC WinCC Unified Alarms – Add a tag to the Alarm Text

The screenshot displays the SIMATIC WinCC Unified Alarms interface. The 'Project tree' on the left shows the project structure, including 'Workshop\_Alarm2', 'HMI\_1 [MTP700 Unified Basic]', and 'HMI tags'. The main window shows the 'Analog alarms' table with one entry: 'Analog alarm\_1' with the text 'Analog Threshold Reached'. A context menu is open over the text, with 'Insert parameter field...' selected. A dialog box for 'Insert parameter field...' is also visible, showing the 'AlarmAnalog' tag selected from a list of HMI tags.

**1** Open Analog alarms

**2** Right click in alarm text

**3** Select 'Insert parameter field...'

**4** Select the AlarmAnalog tag

ID	Name	Alarm text	Alarm class	Trigger tag	Connection of t..	Limit	Limit mod
3	Analog alarm_1	Analog Threshold Reached	Alarm	AlarmAnal...		75	Higher
<Add new>							

Name	Data type	Address
None		
AlarmAnalog	Int	
AlarmByte	Byte	
AlarmSingle	Bool	
RunPB_vlv1	Bool	
RunPB_vlv2	Bool	
Status_vlv1	Int	
Status_vlv2	Int	

# SIMATIC WinCC Unified Alarms – Add a tag to the Alarm Text

1. Save the project

2. Select HMI device in the Project Tree

3. Press Start Simulation, Portal will open the default browser

4. Test your work

5. When finished close the browser

ID	Raise time	Alarm text
1	2/8/2024 8:57	Analog Threshold Reached 76
2		
3		
4		
5		

Alarm Single: 0

Alarm Analog: 76

Alarm Byte: 0000 0000

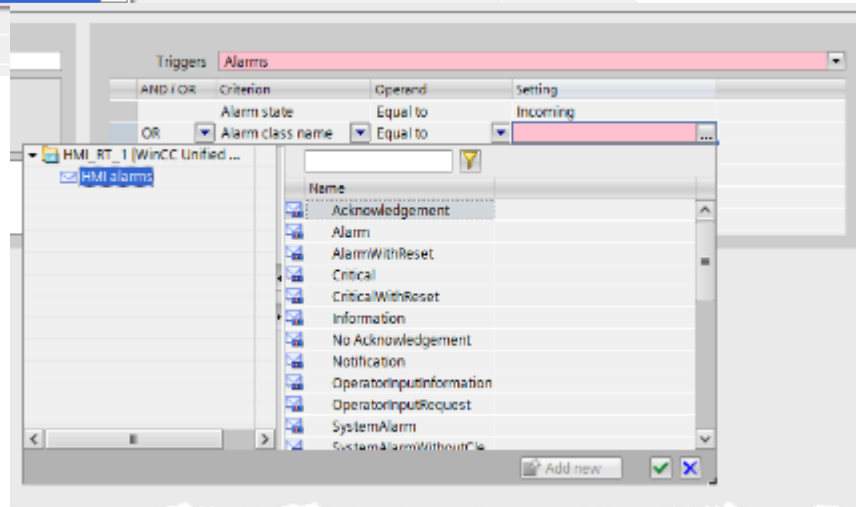
# SIMATIC WinCC Unified Trigger Events by Alarms

## HANDS ON



Unified > Unified [SIMATIC PC station] > HMI\_RT\_1 [WinCC Unified Scada RT] > Scheduled tasks

Name	Trigger	Description	Comment
Task_1	Alarms	Execute as soon as one of the conditions i.	
<Add new>	T500ms		
	T1s		
	T2s		
	T5s		
	T10s		
	Daily		
	Weekly		
	Monthly		
	Yearly		
	Once		
	Tags		
	Alarms		



1. Create a scheduled task
2. Use as trigger “Alarms”
3. Combine some Criteria
4. Trigger the Event during Runtime

# SIMATIC WinCC Unified Trigger Events by Alarms

The screenshot displays the SIMATIC WinCC Unified interface. On the left, the 'Project tree' shows the project structure, with 'HMI\_1 [MTP700 Unified Basic]' selected under 'Workshop\_Alarm2'. A red circle with the number '1' is placed over the 'HMI\_1' folder. The main window shows the 'Discrete alarms' configuration table. A red circle with the number '2' is placed over the 'Warning' dropdown menu in the 'Alarm class' column for ID 2.

ID	Name	Alarm text	Alarm class	Trigger tag	Trigge..	Connection of t..	Acknowledg...	Ackn...	Acknowledg...	Connection stat..
1	Discrete alarm_1	Alarm Single Triggered	Alarm	AlarmSingle	0		<No tag>	0		
2	Discrete alarm_2	Alarm Byte - Bit 1 Triggered	Warning	AlarmByte	1		<No tag>	0		
<Add new>										

1. Open HMI alarms
2. Change Alarm class of ID 2 to 'warning' as shown

# SIMATIC WinCC Unified Trigger Events by Alarms

The screenshot displays the SIMATIC WinCC Unified interface. On the left, the Project tree shows the hierarchy: Workshop\_Alarm2 > HMI\_1 [MTP700 Unified Basic] > Scheduled tasks. A red circle with the number '1' highlights the 'Scheduled tasks' folder. The main area shows a table of tasks with columns for Name, Trigger, Description, and Comment. A red circle with the number '2' highlights the 'Sets Warning Banner' task. Below the table, the 'Sets Warning Banner [Task]' properties window is open, showing the 'General' tab. A red circle with the number '3' highlights the 'Name' field, which contains 'Sets Warning Banner'. A red circle with the number '4' highlights the 'Triggers' section, which is set to 'Alarms' and contains a table of conditions:

AND / OR	Criterion	Operand	Setting
	Alarm state	Equal to	Incoming
AND	Alarm class n...	Equal to	Warning
<Add new>			

1. Open Scheduled tasks
2. Add a new task
3. Rename task as shown
4. Set Trigger type and add conditions as shown



# SIMATIC WinCC Unified Trigger Events by Alarms

The screenshot displays the SIMATIC WinCC Unified interface. At the top, the breadcrumb navigation shows 'Workshop\_Alarm2 > HMI\_1 [MTP700 Unified Basic] > Scheduled tasks'. Below this is a table of scheduled tasks:

Name	Trigger	Description	Comment
5 Sets Warning Banner	Alarms	Execute as soon as one of the conditions...	
5 Resets Warning Banner	Alarms	Execute as soon as one of the conditions...	
<Add new>			

A red circle with the number '1' is positioned over the '<Add new>' button in the table.

Below the table, the 'Resets Warning Banner [Task]' configuration window is open. It has tabs for 'Properties', 'Events', and 'Texts', with 'Properties' selected. Under 'Properties', the 'General' tab is active. The configuration fields are:

- Name:** Resets Warning Banner (indicated by a red circle with '2')
- Description:** Execute as soon as one of the conditions is met
- Comment:** (empty text area)
- Triggers:** Alarms (indicated by a red circle with '3')

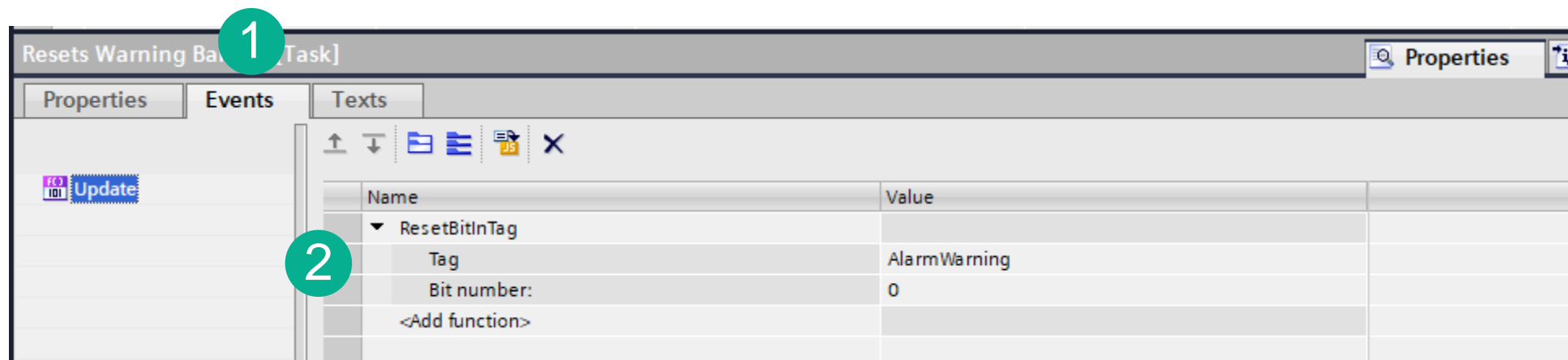
The 'Triggers' section contains a table with the following data:

AND / OR	Criterion	Operand	Setting
	Alarm state	Equal to	Normal
AND	Alarm class name	Equal to	Warning
<Add new>			

1. Add a new task
2. Rename task as shown
3. Set Trigger type and add conditions as shown



# SIMATIC WinCC Unified Trigger Events by Alarms



1. Switch to Events tab
2. Add function 'ResetBitInTag, set value as shown

# SIMATIC WinCC Unified Trigger Events by Alarms

The screenshot displays the SIMATIC WinCC Unified software interface. On the left, the Project tree shows the hierarchy: Workshop\_Alarm2 > HMI\_1 [MTP700 Unified Basic] > Screens > Main Screen. A green circle with the number '1' highlights the 'Main Screen' in the tree. The main workspace shows the 'Main Screen' with a 'Warning!!' text field added. A green circle with the number '2' points to this text field. Below the workspace, the 'Properties' window for 'Text\_1 [Text]' is open. The 'Visibility' property is set to 'Tag', and the tag name is 'AlarmWarning'. A green circle with the number '3' highlights the 'Tag' dropdown menu. The 'Process' section of the Properties window shows the 'Tag' field with 'AlarmWarning' entered. The 'Settings' section has 'Read-' checked. The 'Type' section has 'None' selected. The 'Condition' and 'Visibility' columns are empty in the table below.

1. Open the Main Screen
2. Add a Text field as shown
3. Set visibility to a tag as shown

# SIMATIC WinCC Unified Trigger Events by Alarms

1. Save the project

2. Select HMI device in the Project Tree

3. Press Start Simulation, Portal will open the default browser

4. Test your work

5. When finished close the browser

**Note: The alarm color has changed to Yellow based on the Alarm class settings**

ID	Raise time	Alarm text	
1	2	2/8/2024 10:5	Alarm Byte - Bit 1 Triggered
2			
3			
4			
5			

Alarm Single: 0

Alarm Analog: 0

Alarm Byte: 0000 0010

## WinCC Unified Workshop



### Siemens Industries Inc Digital Industries Factory Automation Visualization

Subject to changes and errors. The information given in this document only contains general descriptions and/or performance features which may not always specifically reflect those described, or which may undergo modification in the course of further development of the products. The requested performance features are binding only when they are expressly agreed upon in the concluded contract.

All product designations, product names, etc. may contain trademarks or other rights of Siemens, its affiliated companies or third parties. Their unauthorized use may infringe the rights of the respective owner.

**[usa.siemens.com](http://usa.siemens.com)**