

Visual Designer tools for FDA 21 CFR Part 11

Introduction

The 21 CFR Part 11 regulations from the FDA (Federal Food, Drug and Cosmetic Act) agency sets forth the criteria under which the agency considers electronic records and electronic signatures to be trustworthy, reliable, and generally equivalent to paper records and handwritten signatures executed on paper.

This document describes Visual Designer software features that allow users to easily configure applications in conformance with the 21 CFR Part 11 regulation using Visual Designer v7.0 or higher as an engineering and run-time tool for Windows XP/Vista/7 and Visual Designer CEView v7.0 as a run-time program for Windows CE.

Comments

General

- The software (SCADA) cannot state that it complies with FDA Part11. The software shall provide the necessary tools to allow a user to create a system (application) that is compliant with FDA Part11.
- The SCADA system should not “force” the user to build an application that is compliant with FDA Part11.
- FDA Part 11 compliance is optional during application development.
- An *Electronic Record* is any data that can be saved as electronic media and retrieved later. An *Electronic Signature* is a specific type of Electronic Record that contains the following information:
 - Timestamp
 - User name
 - Meaning of the signature

A *Digital Signature* is a specific type of Electronic Signature, in which the data is encrypted.

- An *Open System* (such as the World Wide Web or *Web*) requires encryption for electronic reports and for the Electronic Signature (Digital Signature).
- Electronic records are associated with events (such as tag change, load recipe, and so forth), whether the user triggered the event or not. Electronic signatures are associated to actions triggered by the user (such as pressing a button, changing a slider, entering a set-point manually, and so forth).

Electronic Records (Event Logger, Alarms, Reports)

- The Part11 rule does not mention whether the electronic records must be stored in a standard database (such as: Oracle, SQL Server, and so forth) or in a proprietary format. When using a standard database, the responsibility for guaranteeing the confidentiality of the database relies on the database itself (such as password-protected databases).

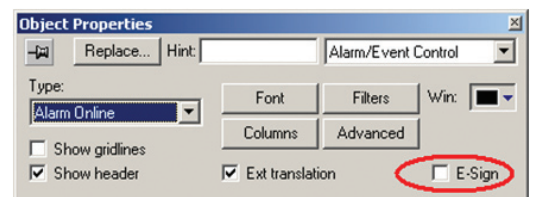
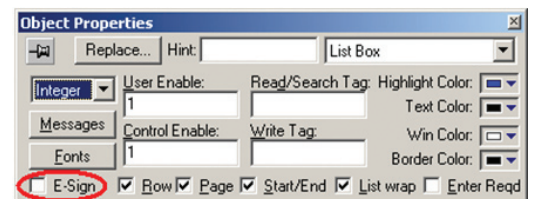
Electronic Signatures (Security System)

- The system administrator must be able to access the user account settings to create new accounts, lockout users, and de-authorize them. These changes must be logged, even if the runtime is not running.
- Nobody (not even the System Administrator) can have access to the password of any user.

Electronic Signatures

This section describes the **E-sign** check-box, which is a property for all objects that enable users' run-time actions. This box requires the user to enter an electronic signature to use certain objects in an application.

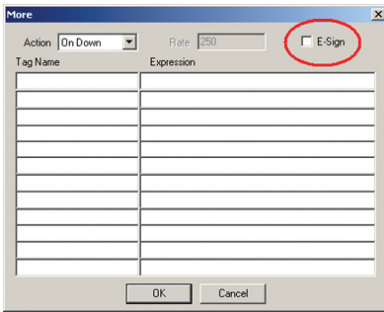
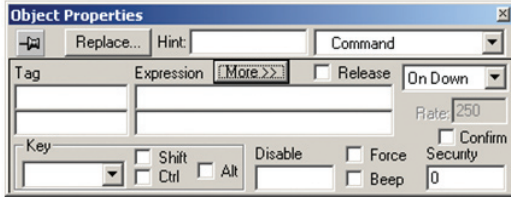
The **E-sign** check-box is part the *Object Properties* dialogs (as shown in the following figures):



EATON

Powering Business Worldwide

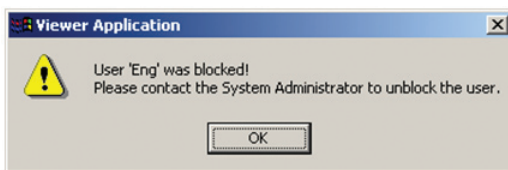
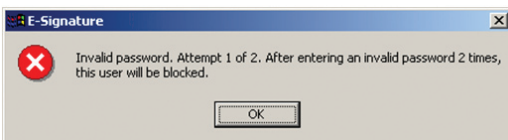
Also, for objects using the **Command** property, the **E-sign** check-box is on the *More* dialog, as shown in the following figures:



If you enable (*check*) this box, users must enter their Electronic Signature before Visual Designer can execute the command. The user must type their **User Name** and **Password** into the following *E-Signature* dialog:



If the user enters an invalid signature, the following message displays and Visual Designer records this event in an Event Log file. For more information about the Event Log file, see the "Event" section in this document.



Instructions for unlocking an account are provided in the next section.

Security System Settings

This section describes the following security system features which address the FDA requirements:

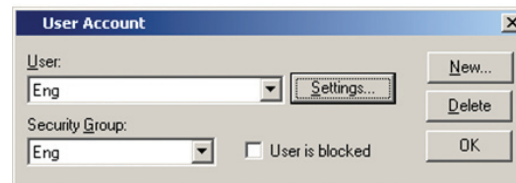
- A **Settings** button appears on the *User Account* dialog
- An **Advanced** button appears on the *Group Account* dialog
- A remote security system can be configured

Settings Button

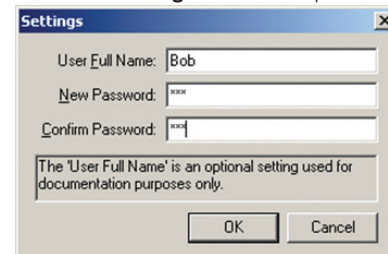
You can use the **User is blocked** check-box and the **Settings** button to control a user's access to the application. You can access these features from the *User Account* dialog.

Use the following steps to open this dialog and configure user access:

1. In the *Workspace*, expand the *Security* folder and right-click on a user name.
2. When the pop-up menu displays, select **Properties** to open the *User Account* dialog.



3. If necessary, click **User is blocked** check-box to block the selected user.
4. Click the **Settings** button to open the *Settings* dialog:



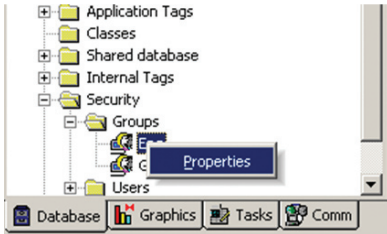
5. Configure the parameters on this dialog as follows
 - **User Full Name** text box (*optional*): Type the user full name.
 - **New Password** text box: Type the user password.
 - **Confirm Password** text box: Re-type the user password.
6. When you are finished, click **OK** to apply the changes and close the *Settings* dialog.

Advanced Button

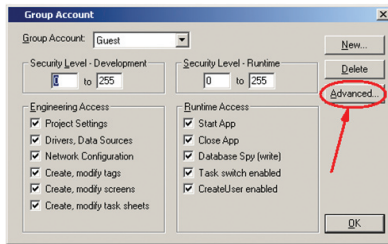
You can use the **Advanced** button to control a user’s access to the application. You can access these features from the *User Account* dialog.

Use the following steps to open this dialog and configure user access:

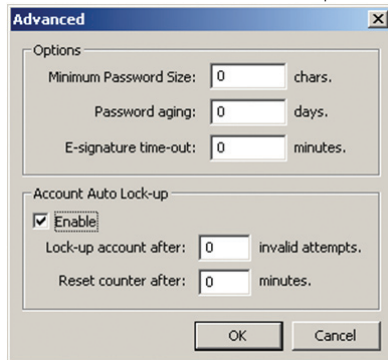
1. In the *Workspace*, expand the *Security* and *Groups* folders, and then right-click on a group name.
2. When the pop-up menu displays, select **Properties** to open the *Group Account* dialog.



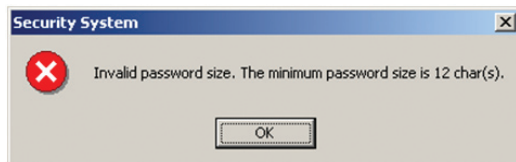
The **Advanced** button appears just below the existing **New** and **Delete** buttons:



3. Click the **Advanced** button to open the *Advanced* dialog:



4. Configure the parameters on the *Advanced* dialog as follows:
 - **Minimum Password Size** text box: Type a value greater than 0 into this field to require a minimum number of characters for a password. All users assigned to this group must provide a password containing at least the minimum number of characters. If a user tries to create a password with less than the required number of characters, Visual Designer will reject the password and display the following warning:



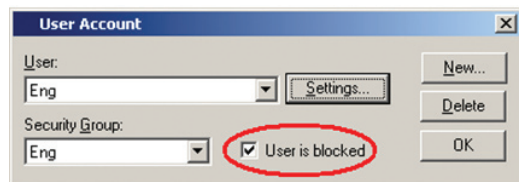
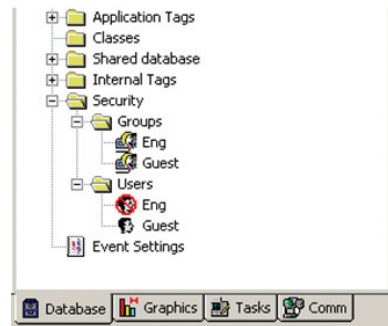
- **Password aging** text box: Type a value greater than 0 into this field to establish the longevity (in days) of a password. After the specified number of days, Visual Designer will force users assigned to this group to change their passwords. When the user tries to log in, the *Change Password* dialog displays (see the following figure) automatically and the user cannot complete the log-in process until they provide a new password.



- **E-signature time-out** text box: Type a value greater than 0 into this field to specify a time-out period (in minutes). Users assigned to this group must enter their UserID and password before the specified timeout period expires to execute commands requiring an electronic signature. Before the time-out time expires, the user is asked for his password only – the system automatically assumes the username logged in the last electronic signature. The system resets the time-out counter just after an electronic signature is executed.
- **Enable** check box: Enable (*check*) this box to activate the following Account Lockup features.
- **Lock-up account after** text box: Type a value into this field to define the maximum number of times a user can try to log on to an account. If the user exceeds the specified a maximum number of attempts (provides an invalid password), Visual Designer will lock the user account.
- **Reset counter after** text box: Type a value into this field to define how long after an invalid log-on attempt, Visual Designer will wait (in minutes) until it resets the log-on attempts counter.

Note: When a user exceeds the specified number of log-on attempts, Visual Designer automatically blocks the account and will not reset the counter — even after the **Reset counter after** time expires. The System Administrator must reset the user account by disabling (*unchecked*) the **User is blocked** check-box on the *User Account* dialog or by executing the UnblockUser() function. (See the “Security System Functions” section.)

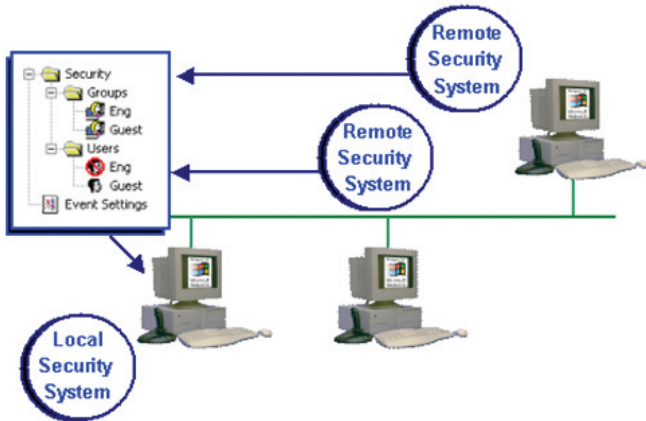
A red circle surrounding a user name in the *Workspace* indicates that the user is blocked. In addition, the **User is blocked** box is enabled (*checked*). For example, the following figures indicate that the *Eng* user is blocked:



- When you are finished, click **OK** to apply your changes and close the *Advanced* dialog.

Remote Security System

If your system applications connect through a TCP/IP link, it is possible to designate one of your computer stations as the *Central* security system, from which other stations can use the Users and Groups definitions. The following figure illustrates this configuration:



Use the following procedure to configure a central security system:

- Right-click the *Security* folder on the **Database** tab and select **Settings** from the pop-up menu to open the *Security System* dialog as follows:



- Enable (check) the **Use preferentially the Remote Security System** check-box to designate a remote security system.
 - If the remote applications successfully connect to the security system from the *Server* station, they will use the security system configured on the *Server* station. In this case, any change implemented in the security system of the *Server* station will be assumed automatically by the remote applications. Also, the security system functions (such as **CreateUser()**, **RemoveUser()**, **ChangePassword()**, and so forth) will update the *Server* station's security system—even if the functions are executed from the remote applications. As a result, all applications on a distributed system can share the same security system settings.

- If the applications cannot connect because the remote system is not running or cannot be reached, a message (similar to the following) will be logged in the *Output* window and saved in the *event* file:

Error connecting to Remote Security Server '192.168.1.255'

In addition, the application(s) will revert to using the local computer's security settings. The remote applications attempt to connect to the *Server* station's security system only when there is an event associated with the security system (such as a user logging on). In other words, there is no polling between the remote applications and the *Server* station during runtime.

Logging and Event Retrieval

This section describes Visual Designer's logging and event-retrieval features. An event can be any tag change, generating reports or recipes, opening and closing screens, logging onto and logging off the security system, and so forth. Visual Designer saves all of these events in a log file, which can be retrieved by the Alarm/Event Control object.

Event log files are stored in the application's *Alarm* folder (the same folder where Visual Designer saves alarm history files). The event log file names must conform to the **evYYMMDD.evt** format, where:

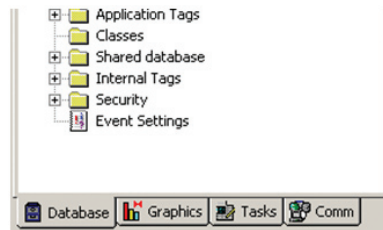
- YY** represents the last two digits of the year in which the event log file was generated
- MM** represents the month in which the event log file was generated
- DD** represents the day on which the event log file was generated

For example, a log file for May 7, 2003 would be **ev030507.evt**.

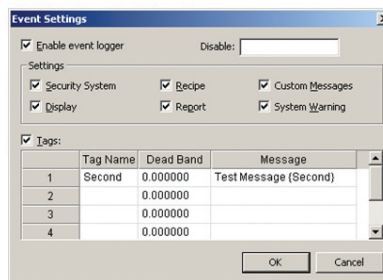
Configuring the Events Settings

Use the following instructions to configure the event-retrieval feature:

- Select the **Database** tab. This tab contains an icon called **Event Settings**, as shown in the following figure:



- Right-click the **Event Settings** icon and select **Open** from the pop-up to open the *Event Settings* dialog.



3. Configure the parameters on the *Event Settings* dialog as follows:

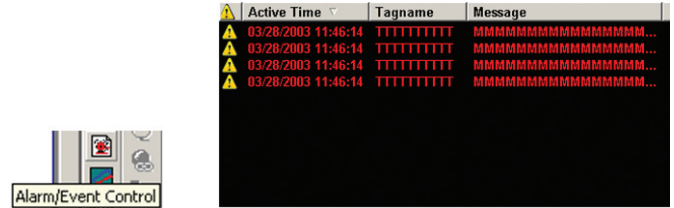
- **Enable event logger** check-box: Enable (*check*) this box to enable event-logging.
- **Disable** text box: Type a tag into this field. If the tag value is other than 0 (*false*), Visual Designer automatically disables the Event Logger.
- **Security System** check-box: Enable (*check*) this box to include security system events in the historic event file. Visual Designer logs the following security system events:
 - Log On / Log Off users
 - User created/removed using the **CreateUser()** or **RemoveUser()** functions
 - User blocked/unblocked using the **BlockUser()** or **UnblockUser()** functions
 - User blocked by the security system after several attempts to enter an invalid password
 - Password expired
 - Password modified
 - Invalid Log On attempt
- **Display** check-box: Enable (*check*) this box to include screen Open and Close events in the historical event file.
- **Recipe** check-box: Enable (*check*) this box to include recipe load, save, init, and delete events in the historical event file.
- **Report** check-box: Enable (*check*) this box to include *reports saved to disk or send to printer* events in the historical event file.
- **Custom Messages** check-box: Enable (*check*) this box to include events generated by the **SendEvent(strEvent)** function in the historical event file.
- **System Warning** check-box: Enable (*check*) this box to include general system warnings (such as *Division by zero*, *Attempted to access invalid array index*, and so forth) in the historical event file. Visual Designer logs the following system warning events:
 - Errors that occur when sending alarms by email
 - Tag was blocked/unblocked
 - Division by zero
 - Connection/Disconnection of the remote security system
- **Tags** check-box: Enable (*check*) this box to enable and log tag changes in the historical event file. Configure the tags you want to log in the Tags table as follows:
 - **Tag Name** column: Type the name of the tag you want to log in the event file.
 - **Dead Band** column: Type a value to filter acceptable changes against.
 - For example, if you specify a **Dead Band** value = 5 for a tag value = 50 and the tag value changes to 52, the system will not register this variation in the event log file, because the variation is less than 5. However, if the tag value change is equal to or greater than 5, the system will save the new value to the history file.
 - **Message** column: Type a string (message) related to this tag change. You can specify tags in messages using the **{tag name}** syntax.

The **Tags** parameter can be useful if you want to generate a log file of events that are not necessarily alarm conditions (for example, **Motor On**, **Motor Off**, and so forth).

Viewing Logged Events

You can use the **Alarm/Event Control** screen object to view a list of all logged events. Use the following instructions to configure this object:

1. Click the **Alarm/Event Control** icon to add an alarm/event control object to your application screen.



2. Double-click on the new object to open an *Object Properties* dialog.
3. Click the **Type** combo-box drop-down arrow and select **Event or Alarm History + Event** from the list.
4. You use most of the same parameters to configure Event objects and Alarm History objects. The only difference is that the **Selection, Group, and Priority filters** are disabled for the Event-type object.

Note: If you are not familiar with the *Alarm/Object Control* screen object, please read the Alarm Control Object section in the "Toolbars" chapter of the *Visual Designer Technical Reference Manual*.

Visual Designer Functions

This section describes the Visual Designer functions associated with the security features discussed previously.

Security Functions

This section describes the following Security functions:

- BlockUser(strUserName)
- UnblockUser(strUserName)
- GetUserState(strUserName)
- SetPassword(strUserName, strOptionalNewPassword)
- CreateUser(strUserName, strGroup, strPassw, strOptUserFullName)
- RemoveUser(strUserName)

BlockUser(strUserName)

Group	Security
Execution	Synchronous
Windows XP/Vista/7	Supported
Windows CE	Supported
Web Thin Client	Supported

- **Description:** Use this function to block an existing user from the security system.
- **Parameters:** This function accepts the following parameter(s):

StrUserName	String tag containing the name of the user to block.
--------------------	--

• **Returned Values:**

0	User blocked successfully
1	Invalid number of parameters
2	Wrong parameter type
3	User does not exist
4	It is not possible to write the data safely

• **Examples:**

Tag Name	Expression
Tag	RemoveUser("Bob")
Tag	RemoveUser("Albert")

Note: You can use this function to remove only those users you created using the CreateUser() function.

Event Logger Functions

This section describes the new **SendEvent(strEvent)** event logger functions:

Group	Event Logger
Execution	Synchronous
Windows XP/Vista/7	Supported
Windows CE	Supported
Web Thin Client	Supported

- **Description:** Use this function to send an event to the Event Log file.
- **Parameters:** This function accepts the following parameter(s):

StrEvent	String tag containing the text to be saved in the Event Log file
-----------------	--

• **Returned Values:**

0	Success
1	Event Logger is disabled in the Event Settings dialog
2	Event Logger is enabled, but Custom Messages are disabled in the Event Settings dialog

• **Examples:**

Tag Name	Expression
Tag	SendEvent("Valve Open")
Tag	SendEvent("Valve Open Oven No." + OvenID)

System Information Function

This section describes the new **SaveAlarmFile(numType, optRemotePath)** system information function.

Group	System Info.
Execution	Synchronous
Windows XP/Vista/7	Supported
Windows CE	Supported
Web Thin Client	Not supported

- **Description:** Use this function to see the current status of a selected user.
- **Parameters:** This function accepts the following parameter(s):

NumType	Tag containing the number and operation, as follows: <ul style="list-style-type: none"> • 0 – Disable save the alarm file to the local disk • 1 – Enable save the alarm file to local disk • 2 – Enable save the alarm file to local disk and to the remote path specified in the OptRemotePath parameter
OptRemotePath	Tag containing the name of the remote computer where the alarm file will be saved simultaneously to the local computer and to the remote path when numType = 2 .

• **Returned Values:**

0	Success
1	2nd parameter is not a string
2	2nd parameter is missing

• **Examples:**

Tag Name	Expression
Tag	SaveAlarmFile(0)
Tag	SaveAlarmFile(1)
Tag	SaveAlarmFile(2, "Z:\Apps\AppDemo")

Eaton Corporation
Electrical Sector
1111 Superior Ave.
Cleveland, OH 44114
United States
877-ETN-CARE (877-386-2273)
Eaton.com

© 2011 Eaton Corporation
All Rights Reserved
Printed in USA
Publication No. AP04803009E / TN
May 2011



Eaton is a registered trademark of Eaton Corporation.

All other trademarks are property of their respective owners.