Visio Logger

Getting the most from the Visio Logger system

Table of Contents

What Is Visio Logger?

Visio Logger is a *Generalized* Data Collection System

One of the powerful features of Visio Logger is its *configurability*. The system assumes nothing about what information you may want to gather. Because of this, you can put it into service in any environment, out of the box, without any constraint such as those imposed by the software design groups of our competitors.

More than Facilities Sign In/Out System

Visio Logger was initially developed to serve as a powerful lobby, or facility, visitor logging tool. However, the configurable nature of the system, coupled with its highly extensible *architecture,* means that it can be used to explore new concepts in visitor **or** data management – well beyond simple logging of visitors within a facility.

Flexible and Configurable Data Definitions

Visio Logger works by providing you *containers* within which you define the data that you need to collect. Internally, Visio Logger is built to deal with the Metadata of your data, which is a fancy way of saying "Data about Data". This means that the system is completely open to the types and characteristics of whatever data you want to define.

Using Visio Logger's unique document storage fields, you can even create a complete document management system with the system to manage any type of document on a per visitor basis.

Flexible and Configurable Input Mechanisms

Visio Logger can accept input from a standard electronic signature pad – *either* alphanumeric data entered by the signer on a *keyboard* displayed on the pad; by presenting a menu of options to the signer on the pad; or, handwriting from the pad, which can be more than just a signature.

The system can *also* be used as a pure Kiosk system, where visitors have access to the keyboard, monitor, and mouse, and you can present to them a professional and highly customizable graphical experience. With a touch screen monitor – your visitors need not have access to a keyboard *or* mouse.

Additionally, switching between Kiosk and signature pad entry mode is a simple menu option. These modes are completely interchangeable and your data definition(s) will work seamlessly in either mode.

Further – you can have a Kiosk system *with* a signature pad and can gather a signature for every visitor, seamlessly.

Activity Management

Visio Logger is also a *time* and *activity* management system. All sign in and out systems would have the built in time stamps for each visitor. However, this system goes well beyond that and allows *you* to define any action for your visitors[1].

An important part of the system is that it is intended to be used throughout an enterprise for managing visitor data. When Visio Logger is running on any computer in Viewer Mode[2], any other computer *can* act upon the visitor data through Actions that are defined and configured *by you*, the end user of the system.

It is much more than the fact that any computer can *see* the visitor's data in real time. Now, you can build up the data about your visitors' total experience within your facility and start to track activity, including time and location, for any number of activities.

The Action fields that you create also have built in *alerting* mechanisms which allow all computers on the network to immediately see when a defined "time exceeded" condition has been reached. These alerts may also be used as indicators that the visitor has supplied a given value that might be important for the facility to be aware of. Note also that the "time elapsed" alerts may *also* be driven by a certain value having been provided.

As a specific example, if you have an action called "Patient seen", and one of your menu driven data entry fields, perhaps "Symptom", has a value "Chest Pain", then – you can set up the alerting mechanism such that the row of data turns yellow after a configurable amount of time, then red after another amount of time, and so on. Until the "Patient seen" action is taken by some computer on the network – this alerting mechanism will continue to monitor that record for these alerts, even if Visio Logger is shutdown and restarted. Thus, unhandled alert conditions will always show the alert status.

Easy Integration with other systems

Visio Logger has a powerful integration architecture that allows a high level of extensibility within itself, and within an organization's other processes and procedures.

At 6 different points within the visitor data collection process, the system is able to pass information back and forth to external tools, and allows those tools to consume the data collected so far, *as well* as to *set* the remaining data for any reason.

As examples of this, several powerful tools are provided with Visio Logger. For some of these, the source code is provided for use by organizations who would like to create their own integration tools.

The tools provided with Visio Logger include:

- Page Submitted Processor: This tool provides the capability to setup multiple "paths" in the visitor sign in process based on responses the visitor has provided. The tool has configuration settings that allow you to set the value of fields that would occur later in the process based on

[1] We use the term "visitors" to refer to the main resource associated with a given row of data. However, this could easily be a "vehicle" in the motor pool, an item on an assembly line, or a tool from the tool shed.

[2] Please see Page 7

the value of fields just entered or previously entered by the signer. Because Visio Logger skips fields that already have a value, this effectively configures the system to present a different chain of entry fields to the signer based on any or multiple previously entered values.

To see more about this tool, peruse the "Using the InnoVisioNate Page Submitted Processor" document from the InnoVisioNate website (www.innovisionate.com)

- Sign Out Processor: This tool allows you to turn your Sign In processor into a complete stand-alone Sign In AND Out system. The configuration options allow you to indicate which field, or combination of fields, in your data that will be presented to a signer at Sign Out time to allow the system to *find* the Sign In record and how to mark that record as "Signed Out".

  To see more about this tool, find the document "Using the InnoVisioNate Sign Out Processor" from the InnoVisioNate website.

- Badge Printing Processor: After any visitor has signed in, you can use this tool to print a badge for that visitor on any badge printer of your choice.

  The configuration of this tool allows you to layout your chosen fields onto a template that is the same size as your badge media. You can embellish the layout with any additional text need, and you can even display signatures, or any image data (see the Video Processor description for more information).

  To see more about this tool, find the document "Using the InnoVisioNate Badge Printer" from the InnoVisioNate website.

- Video Processor: This tool seamlessly provides the capability to capture a picture of the visitor, or any object, during sign in time, or, any time in the future, from any computer with Visio Logger and some sort of webcam.

  The source for this tool is provided with Visio Logger and with that, any organization can see how to create these integration tools for other purposes. For example, with the appropriate hardware and software interface, this source code could provide the foundation to implement a feature to unlock a door, given certain information provided by the visitor passes some sort of validation, which, of course, you can perform right in the integration tool.

  This tool is described in this document, please see InnoVisioNate Video Tool on Page 60.

- Spreadsheet Processor: You can automatically populate any Excel Workbook with data coming from your visitors in real time. This tool allows you to configure all or a subset of your fields to be placed in a worksheet for every visitor. It can also find this data in the worksheet and update the appropriate row when the data for a visitor changes, such as when an action is taken, or the person signs out.

Any individual in an organization can see a live snapshot of all visitor data for as far back as the first use of this tool.

The source code for this tool is also included with Visio Logger. An entity could easily enhance the tool to automatically create a new workbook every day, for example.

For more information, please reference InnoVisioNate Spreadsheet Tools on Page 60.

You can start doing all of this tomorrow – Out of the box with Visio Logger.

Using Visio Logger – Station Type

Remember that Visio Logger is a data collection *and* management tool. When we talk about data collection, we mean Visio Logger is running with a signature pad connected to the computer (or is running in Kiosk Mode), and visitors are using that computer[3] to sign in to (or out of) the facility.

For all modes of operation, it is assumed that every computer accesses the same database so that they are all referring to the same visitor records.

Visio Logger will remember the mode it was in whenever it is restarted. Therefore, if the last time the system was closed it was in Logger mode, when it starts, it will automatically begin gathering data again.

Additionally, upon installation, Visio Logger can be setup to start automatically when the computer starts (after any required user login) in whichever mode the individual installing the software chooses.

This way, the system can restart collecting data in the event of a power outage in an unmanned location.

Logger Mode

To start collecting data, choose Edit → Station Type → Logger from the main menu. Note that this option *is* available even if the user does not have administrative privileges. Therefore, staff can switch between gathering data and simply viewing it by changing the station type.

Note that facilities with multiple licenses for running Visio Logger in Logger mode *can* be collecting visitor data from more than one location, from all entry doors, for example. There is no requirement that only one station be used at a time.

[3] With the exception of Kiosk mode – visitors do not, however, see the monitor, keyboard or mouse of the computer.

Logger Mode also *contains* Viewer Mode, in that the rows of data are always visible and can be acted upon just as in Viewer Mode. In Logger Mode, however, an additional window is always up which represents the signature pad and what the signer is seeing at every point in time.

Viewer Mode

Any staff in the facility, or in any facility with network access to the database the computer using Visio Logger in Auto Logger mode is connected to, can watch the collected visitor data in real time.

These computers would be running the system in “Viewer Mode”. To enter this mode, choose Edit → Station Type → Viewer from the main menu.

Viewer mode is also intended to be used on computers where actions related to the visitor may be taken. An example might be when the visitor visits a certain station or office in the facility, and you want it recorded when they were there. Any staff at that station or office would simply find that visitor in the current day’s logging activity, and click the appropriate button for the visitor.

Additionally, in Viewer Mode (in any mode except Kiosk, for that matter), users at the computer *can* edit cells which are marked as Editable in the fields definition. For example, to provide notes or any clarifying information about the visitor.

Auto Logger Mode

Visio Logger can be used as a complete Sign In *and* Out system simply by choosing this mode, accessed via Edit →Station Type → Auto Logger.

This mode is intended for totally unmanned operation, where staff is generally not present and the data collection needs are relatively simple. Specifically, when there is only one “path” through the signer data.

When we say “path”, we mean that in more complex configurations, Visio Logger *can* present a different series of pages to the signer based on prior answers they have provided. To see a full description of this, please refer to the “Using the InnoVisioNate Page Submitted Processor” document.

Therefore, in the absence of multiple paths, which Auto Logger Mode does not implement, and in the presence of at least one Action field, after performing a simple configuration step, you can immediately turn your sign IN system into a stand-alone and fully functional Sign In *and* Out system by using Auto Logger mode.

Because Auto Logger mode requires a setup and configuration step, we have dedicated a section to describing this after we provide information on Pages and Fields. Please see Configuring Auto Logger operation on Page 34 for more information.

Kiosk Mode

Kiosk mode is a special type of “client” operation where the Visio Logger user interface does not show at all. Instead, a richly configurable set of bitmaps is presented on the computer monitor along with graphical menus and entry fields that a visitor would enter the data in through the hosting computer's keyboard and mouse. Or, with the use of a touch screen monitor, through that monitor alone.

Kiosk mode is described in full in its own documentation. Please find the document "Configuring Visio Logger as a Full Featured Kiosk.pdf" either in the Visio Logger ProgramData directory, or on the InnoVisioNate website.

Using Visio Logger – Creating your data definitions

We used the term "containers" as a place for you to create the fields that you need to define for your facility. This section describes how to fill in those containers to create the "Columns" in your database.

On initial installation, your Visio Logger database[4] will only contain a "Sequence" and a "Signed In" field. The first step is to create a few fields for the data that you want to collect.

You can change, remove, or add fields at any time in the future as your needs evolve.

Using a Signature Pad?

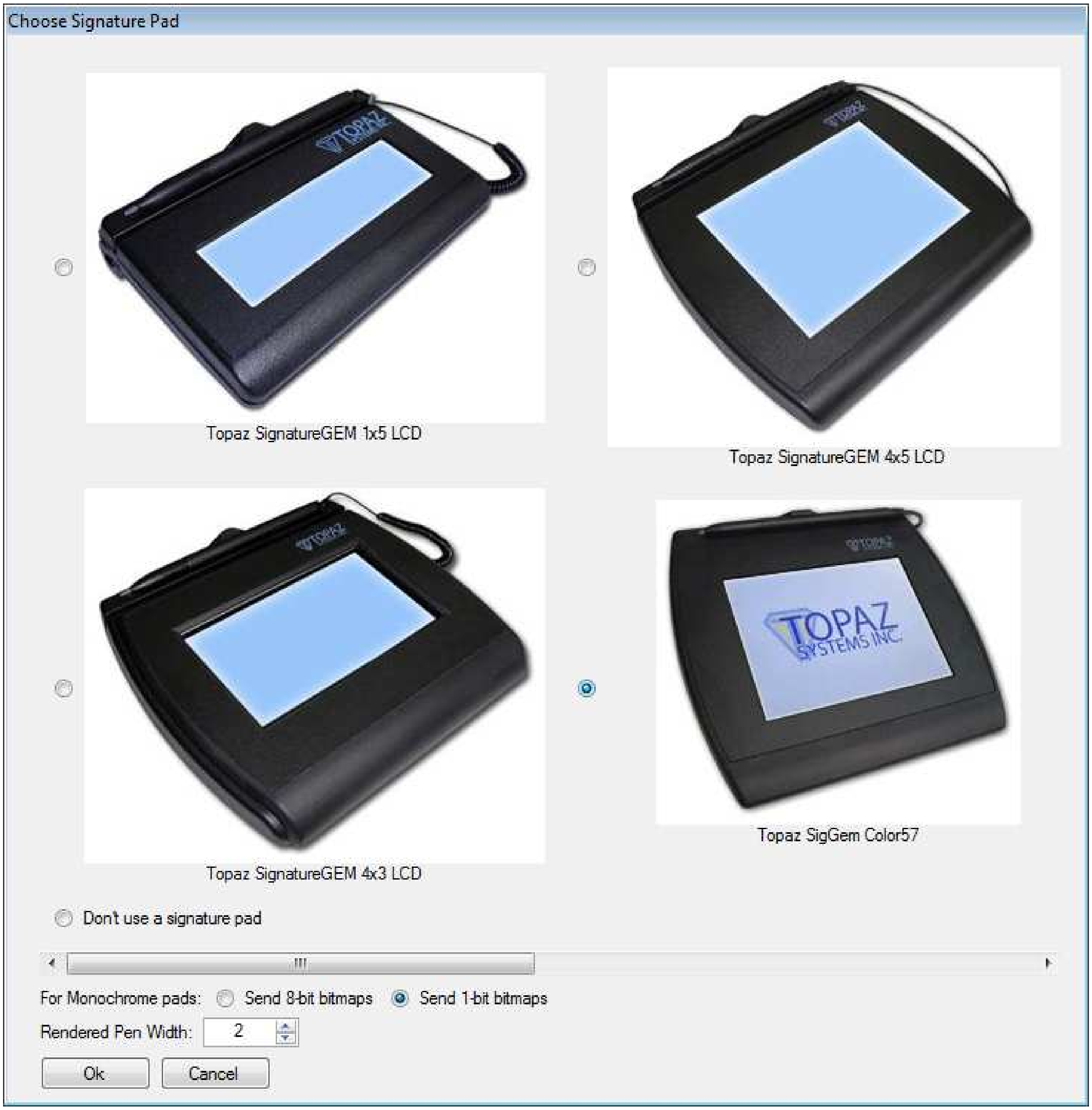
If you will be running Visio Logger with an electronic signature pad, select the pad from the available pads before creating your data columns (described below). This is necessary because within the definitions of your data, you will be specifying whether or not to use the pad, and the system needs to know which pad will be in use at this point.

Note: You can *always* change the pad you want to use, for example, if you want to switch to a larger pad in the future.

Figure 1 Shows pads currently available for use with Visio Logger[5].

[4] Visio Logger can use 4 different types of database systems and can seamlessly switch between them. Please see The Visio Logger Database on Page 40 for more information.

[5] Note that the SignatureGEM 1x5 LCD is too small to display a data entry keyboard, therefore it is not suitable for alphanumeric fields unless they are all "Menus".



Choose Signature Pad

○ Topaz SignatureGEM 1x5 LCD

○ Topaz SignatureGEM 4x5 LCD

○ Topaz SignatureGEM 4x3 LCD

● Topaz SigGem Color57

○ Don't use a signature pad

For Monochrome pads: ○ Send 8-bit bitmaps ● Send 1-bit bitmaps

Rendered Pen Width: 2

Figure 1 Choosing the Signature Pad

In the above figure, we have selected the SigGEM Color57 signature pad. Note that this pad has a resolution of 640x480 pixels and is simply the best pad for use with Visio Logger due to the high clarity and crispness of any text it displays. The 4x5 LCD also works, however, at 320x240 pixels, it is actually ¼ the resolution of the color pad.

For a monochrome pad, choose whether you want to send 8-, or 1-bit images to it. For non-complex images, 1-bit offers better performance.

Finally, you can set the pen width used to create the handwriting bitmaps for any signature fields.

After you have specified your signature pad, you can …

Create Data Pages and Columns

While running Visio Logger as a user with administrative privileges[6], the main menu will contain the Edit → Columns entry. Choosing this option we see the dialog shown in Figure 2.

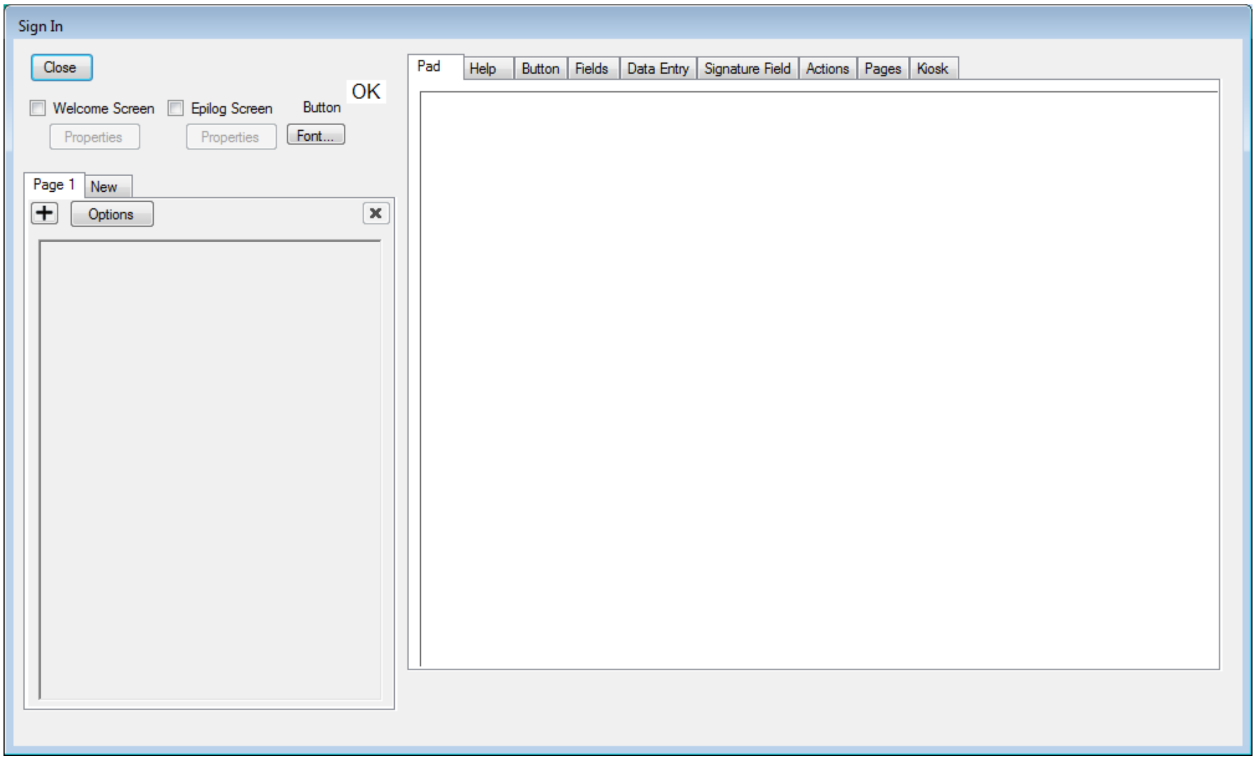


Figure 2 Edit Columns - No Fields Defined

Note that the tab labeled "Pad" is the same size as your signature pad, in our case, 640x480 pixels. As you build up your fields – this tab will contain the exact image of the information displayed on the physical signature pad. Because this pad is relatively large, you may need a larger monitor to work with while creating fields (when Visio Logger is running, however, monitor size has no impact).

What is a "Page"?

You can think of a page of data as *one* presentation of one or more *prompts* to your visitor, *and* the values of the fields gathered from the visitor after they submitted the entries.

In almost every case, a "page" contains only 1 field of visitor entered data. That is *if* the data is entered by the visitor using *either* a keyboard entry, or as a menu option (more about these later).

[6] Most Visio Logger configuration options require the user to have Administrative Privileges. If in doubt, right click on the Visio Logger Icon, and choose Run As Administrator.

However, you *can* include multiple handwriting fields on a single page. When you define handwriting fields, you can choose any region of the signature pad to take the handwriting from, effectively splitting the pad into any number of fields.

For our purposes, we will be creating mostly alphanumeric fields (we want *digital* data, that is, computer readable data, in our database). Therefore, we will be creating a keyboard or a menu selection on the signature pad for the visitor to manipulate for most of our input.

In terms of Figure 2, note that there is one Page defined, and it is empty, that is, it has no data entry fields.

Page Options

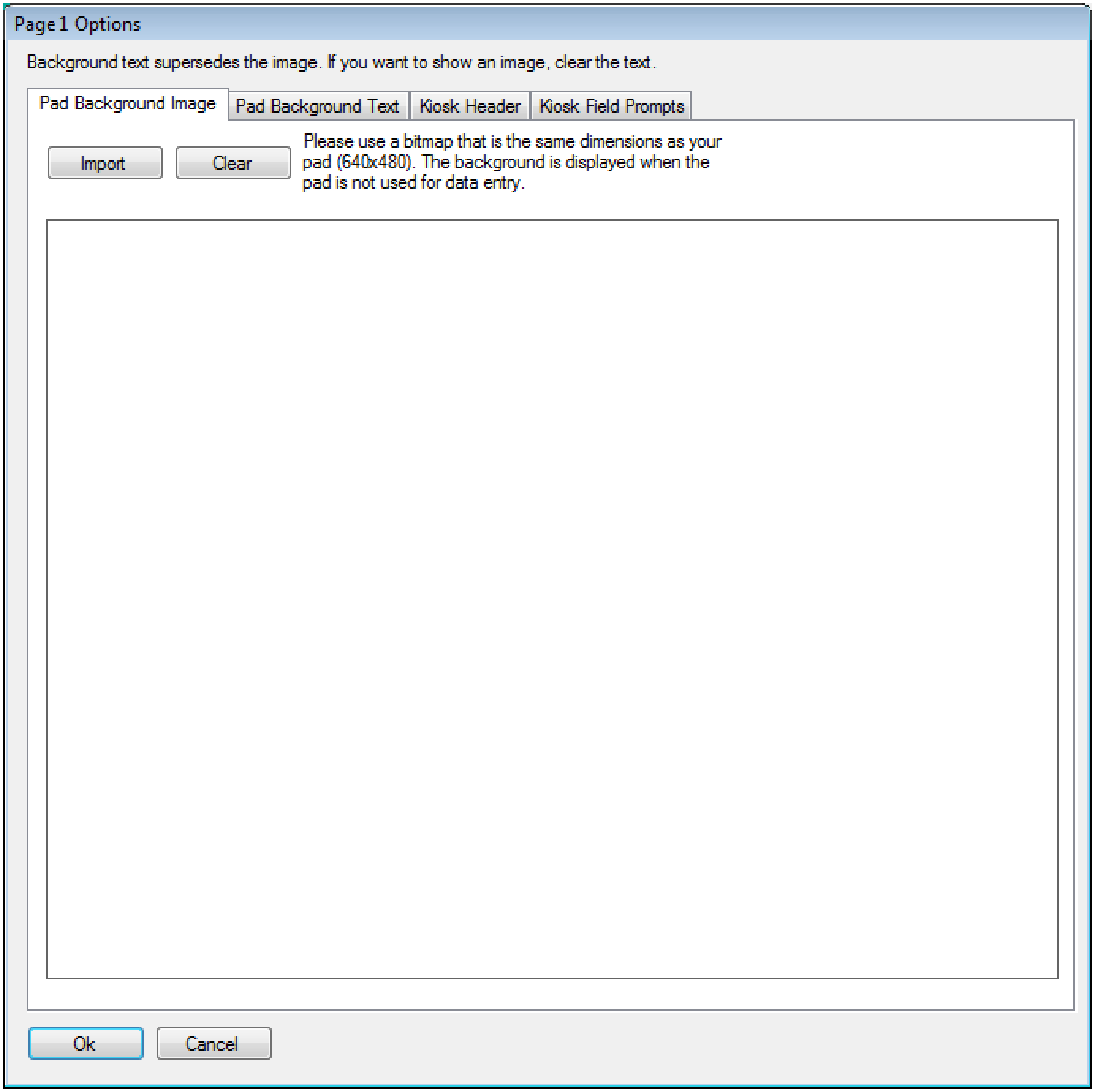
Clicking the Options button beneath the "Page 1" tab, we see Figure 3. These are the options for that Page and are used to send an image or text to the signature pad when that page becomes active (is presented to the signer).

Note the Kiosk Header and Kiosk Field Prompts are relevant when Visio Logger is running in Kiosk mode, and are fully described in the "Configuring Visio Logger as a Full Featured Kiosk.pdf" document.

You can use these options to provide clarifying or additional information to the signer related to the field(s) you are collecting ***when*** collecting handwriting only. The background on the pad is obscured when a keyboard or menu is displayed upon it. Therefore, these options are only applicable to handwriting fields.

However, this option is very convenient for those fields. For example, you could create an "Address Form" with the Pad Background Text option – and split the resulting handwriting into each component of the address, with the form being the guide for the signer.

Note that, when either a background image or text is specified, the *text* specified in the signature field definition is *not* sent to the pad.

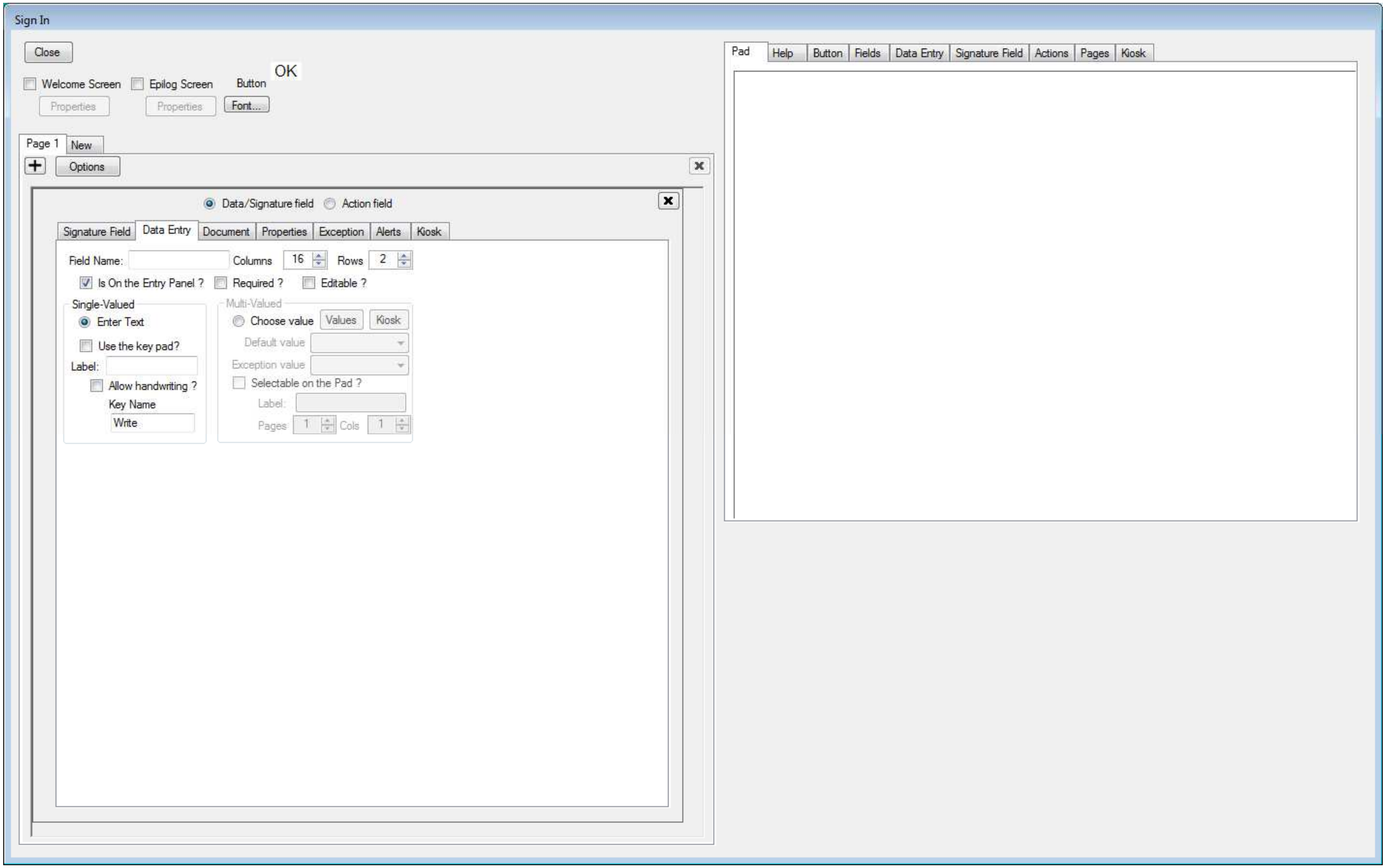


Background text supersedes the image. If you want to show an image, clear the text.

Please use a bitmap that is the same dimensions as your pad (640x480). The background is displayed when the pad is not used for data entry.

Figure 3 Page Options

If we click the ⊞ button directly under the “Page 1” tab in Figure 2, we will create a new field for that page and will see the dialog as in Figure 4..



□ Welcome Screen □ Epilog Screen Button

● Data/Signature field ○ Action field

Field Name: Columns 16 Rows 2

☑ Is On the Entry Panel ? □ Required ? □ Editable ?

Single-Valued

● Enter Text

□ Use the key pad?

Label:

□ Allow handwriting ?

Key Name: Write

Multi-Valued

○ Choose value

Default value

Exception value

□ Selectable on the Pad ?

Label:

Pages 1 Cols 1

Figure 4 Creating our First Data Field

Data Entry Fields

The above figure shows the container into which we can now create our first field.

In this case, we will create an *alphanumeric* field. The distinction is that this "type" of field is "Data Entry". Note that this is the Tab currently selected for the field.

An alphanumeric field is one where the value is digital and therefore useful for later analysis, filtering, sorting, etc. Recall that Handwriting fields are bitmaps. Visio Logger is not able to discern the "value" of handwritten fields, it can only store and display them.

Signature Pad Keyboard Fields

The steps to create a Keyboard Data Entry field are:

1. First, we enter a name for the field. This will be the column header in our rows of data. It can contain any number letter or spaces, but should not contain special characters. We'll enter the name "First Name".

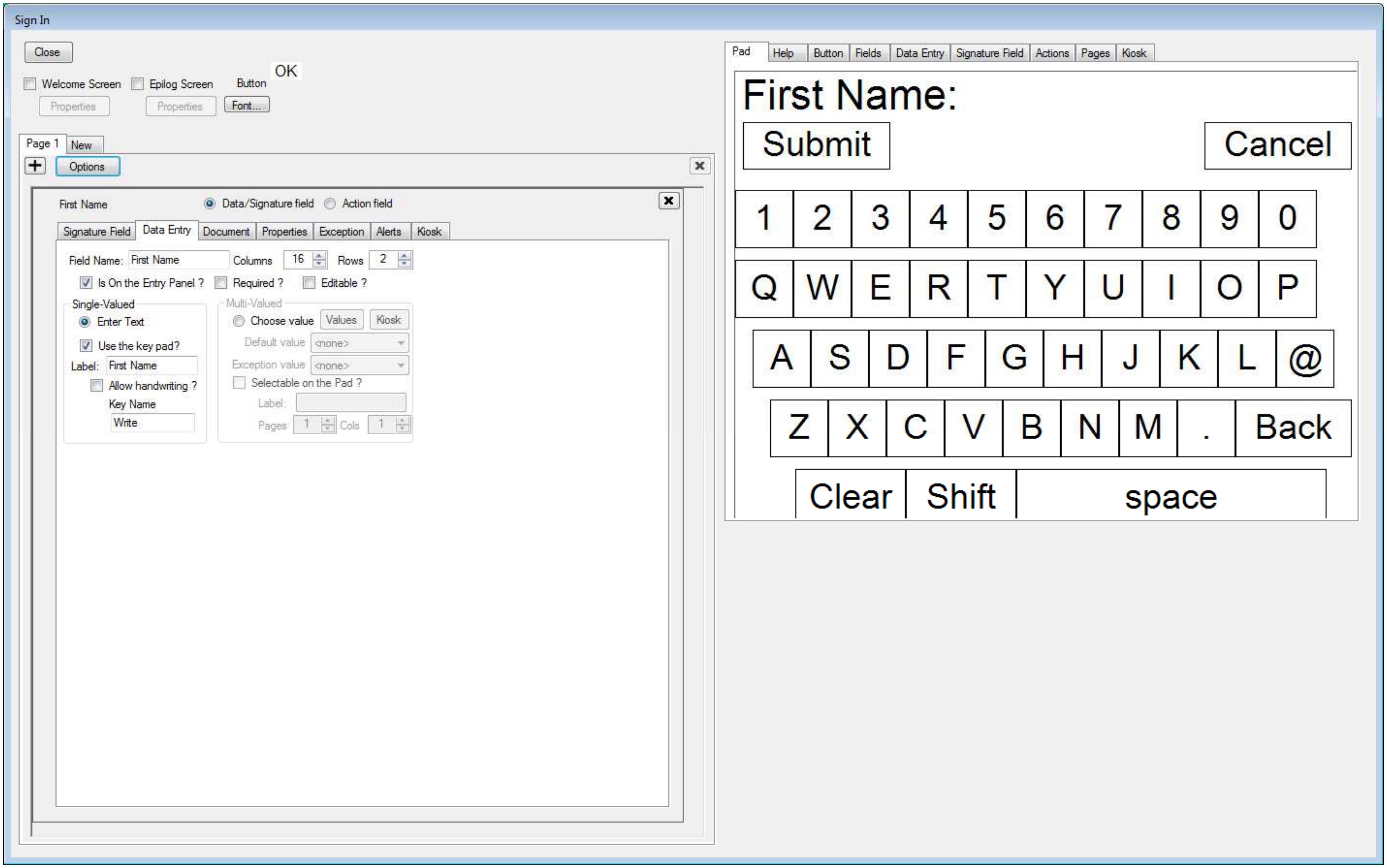
2. Then we ensure "Is On the Entry Panel ?" is checked if this field is to be presented to the signer.

   Note: Fields do not **have** to be on the entry panel. If they are it means is that this is a field that *is* presented to the signer and they will provide a value for. There may be very good reasons to define fields that are not filled in by the signer. A field can be editable, for example, and any staff using Visio Logger on any computer *could* provide value(s) for the field later. An example being a "Notes" field.

3. We choose the display width and height within the Visio Logger "view" of the cell. Note that the height of each Visio Logger row of data will be the largest of these "Rows" values. Except that, if any signature fields are defined, they would typically dictate the height.

4. We select either "Single-Value" or "Multi-Valued" entry type. Note carefully, that to choose "Multi-Valued" – we would click the "Choose value" radio button in the "Multi-Valued" group, which is disabled until "Choose value" is clicked on. In our case, we've selected "Enter Text" *and* checked on the "Use the key pad?" option which means that the signer will be able to tap out their response on a keyboard displayed on the signature pad. Recall that the image on the right side of our field definition dialog is the exact image that will be shown on the pad.

5. Finally, specify a prompt for the data field that will be shown to the signer. This is in the "Label:" field where we have entered "First Name"



□ Welcome Screen □ Epilog Screen

First Name ● Data/Signature field ○ Action field

Field Name: First Name Columns 16 Rows 2

☑ Is On the Entry Panel ? □ Required ? □ Editable ?

Single-Valued

● Enter Text

☑ Use the key pad?

Label: First Name

□ Allow handwriting ?

Key Name Write

Multi-Valued

○ Choose value

Default value <none>

Exception value <none>

□ Selectable on the Pad ?

Label:

Pages 1 Cols 1

Figure 5 Defining the First Name Field

Figure 5 shows our field so far. The right side of the figure depicts what will appear on the signature pad. The signer will simply tap out their first name, tap Submit with the stylus, and Visio Logger will proceed to the next page, if defined, or prepare for the next signer, if this were the last page.

Other options in Figure 5 not yet described are:

- If a column is “Required ?” – the system will not accept the page of data until a value has been provided. This typically only applies to pad Keyboard fields because there is no way to “submit” a menu field, other than to select one of the items. When using the pad keyboard, if a user taps “Submit” with the stylus without having tapped in a value, the pad flashes “Field Name is required” and stays within that field.
- If a column is “Editable ?”, any user on a computer running Visio Logger can click in the cell and change the value, such as for a “Notes” field.
- The Properties tab allows you to set the color of the background, and the text, in the cell for the column. You can have different colors for the “normal” display, or for the display *after* the value has changed for any reason. If the column is “Editable ?”, you can tell if a field has been edited by any user by specifying a different color for the “changed” option.
- The field can be deleted by pressing the ☒button in the upper right.
- The Kiosk tab sets properties useful only to Kiosk mode of operation. Please reference “Configuring Visio Logger as a Full Featured Kiosk.pdf” for detailed information on using this mode.

Allow Handwriting

The “Allow Handwriting” option allows your signers to use the pen and write their response to the field. When this option is on – the system adds an additional “key” to the keyboard, which will appear to the right of the “Submit” key shown in the right of Figure 5. That key will have the value (label) that is shown in the “Key Name” field.

To use this feature – you must define a Signature field on a new Page somewhere. When you define a Signature field, there is an option to set that field as an “Alternative To”. Therefore, to turn on “Allow Handwriting” for a field, Field A, for example, create a Signature field on another page, and select “Field A” as the “Alternative to” in the settings for the Signature field.

When this option is used by the signer – there is NO value in Field A, and that related signature field will contain their handwriting for the value. Typically, you would make Field A editable so at a later time, someone could peruse the hand written values, and fill in the values in Field A.

Recall that handwriting fields are not decipherable as to their contents by Visio Logger. The system makes no attempt to “read” handwriting. The basic premise of this technique is to allow a way for people unfamiliar with keyboards the option to bypass it and write their response.

Menu Fields

Next, we’ll create another Data Entry Field – except this time we will make it a menu field. Note that menu fields allow you to capture “choices” – without the fear of the signer misspelling anything. In other words, with a menu field, you always know that the value will exactly one of a predetermined set of values, and that you can use this to advantage later on.

To create our new field, we *must* put it on a separate page. Recall that Visio Logger can only display one keyboard or menu field on a given page. Therefore, we click the “New” tab, that is just to the right of the “Page 1” tab in the upper portion of Figure 5. We see again the dialog shown in Figure 2, with the exception that “Page 2” is now the “current” page, and that it is to the right of the “Page 1” tab, and that there is a new “New” tab.

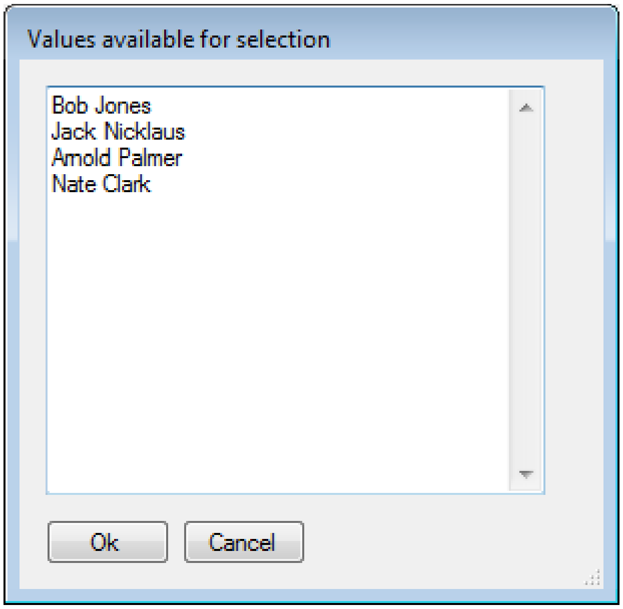
Again, click the ⊞ under the “Page 2” tab header and we’ll work again with the Data Entry tab as shown in Figure 5.

Steps to create a menu field are:

1. Specify the field name for the “Field Name:” value. Here we’ll use “Who To See ?”

2. Specify “Multi-Valued” by clicking the “Choose value” radio button.

3. Provide the available values for the menu. Clicking the “Values” button shows the dialog shown here:



Values available for selection

Bob Jones
Jack Nicklaus
Arnold Palmer
Nate Clark

Figure 6 Specifying Menu options

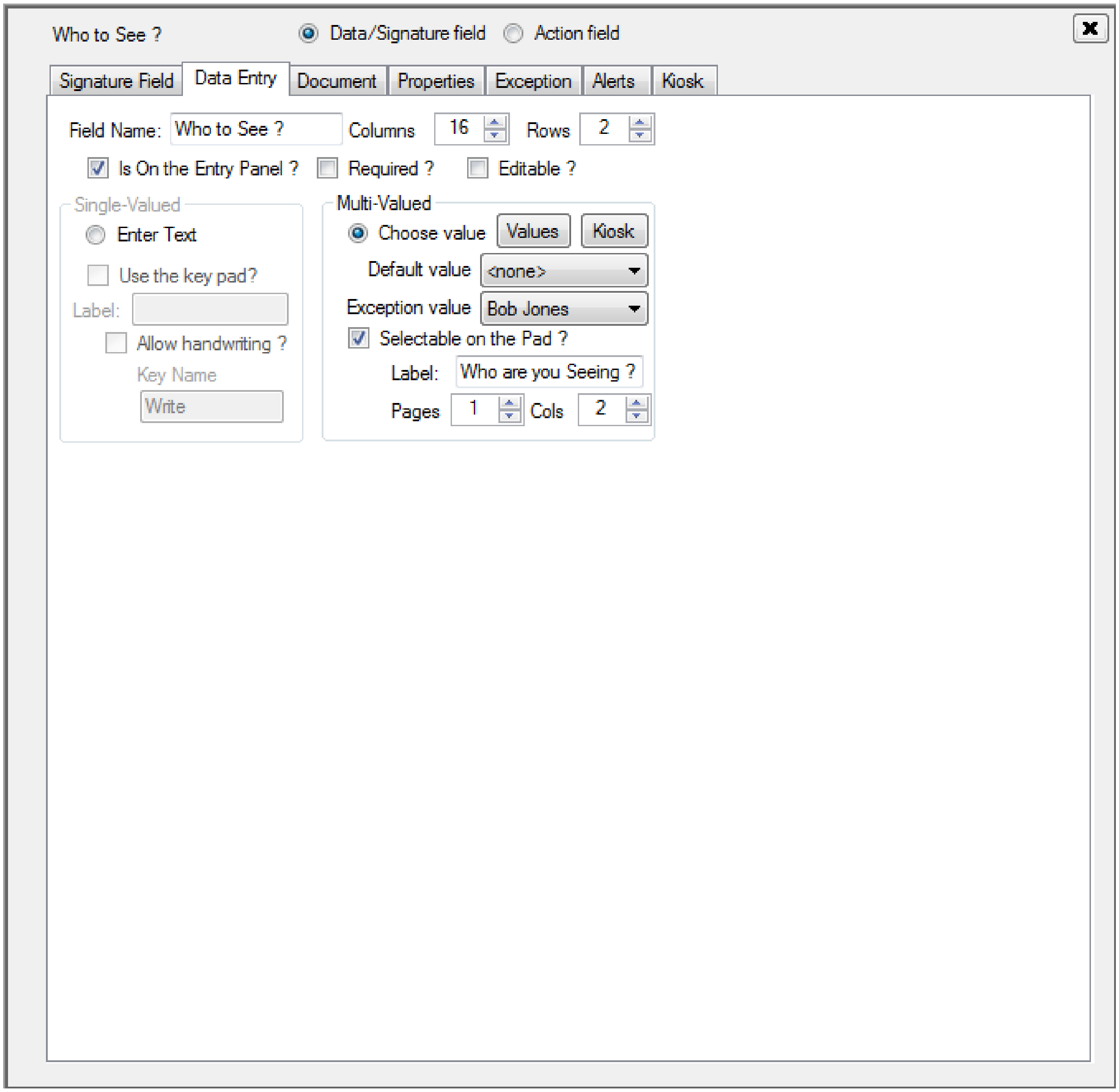
Note in Figure 6 that we have entered the names of several famous golfers for selection by the visitor. We know that our data will *always* contain one of these values because the signer is not able to specify anything else[7].

4. Ensure that “Selectable on the Pad ?” is on. Without this setting, the visitor would not be able to make the selection, and the receptionist *would* have to make it for them. Recall that this may be the desired effect, having reception or other staff provide part of the data while the visitor enters the rest.

5. Finally, set a label (displays at the top of the pad as a prompt), and the # of pages and columns to break the selection into. Note that there is no limit to the available values (you can paste a column of data from an Excel Workbook into the “available” values dialog as in Figure 6). Typically, the large color signature pad can display about 12 rows of 2 columns (24) values. If you have more than one page of data, the system will display a “Next” and/or “Previous” button on the signature pad as appropriate.

Figure 7 shows our completed menu field.

[7] It is very possible to have an “Other” option available, and to prompt the signer with the value of that Other option using an additional field. Please see the documentation on “Using the InnoVisioNate Page Submitted Processor” to see how this is done.



Who to See ? ● Data/Signature field ○ Action field

Signature Field | Data Entry | Document | Properties | Exception | Alerts | Kiosk

Field Name: Who to See ? Columns 16 Rows 2

☑ Is On the Entry Panel ? □ Required ? □ Editable ?

Single-Valued
○ Enter Text
□ Use the key pad?
Label:
□ Allow handwriting ?
Key Name
Write

Multi-Valued
● Choose value Values Kiosk
Default value <none>
Exception value Bob Jones
☑ Selectable on the Pad ?
Label: Who are you Seeing ?
Pages 1 Cols 2

Figure 7 Fully defined Menu Field

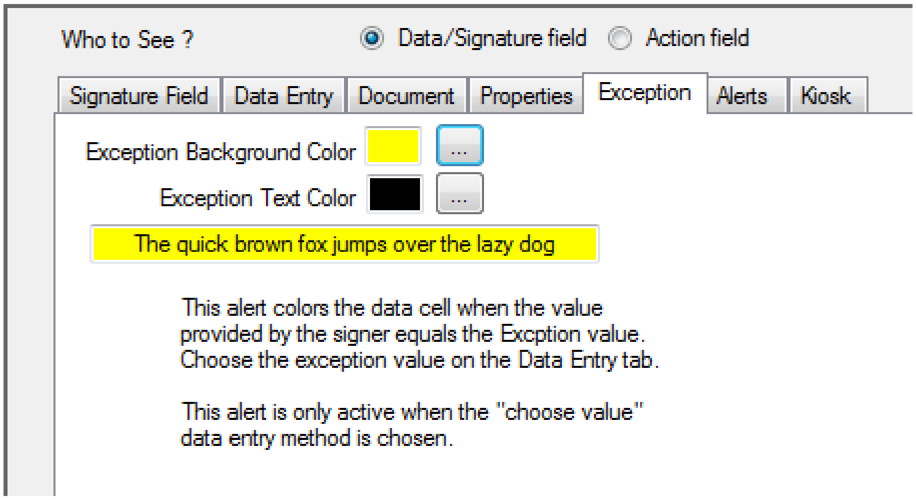
Using the menu driven alerting mechanism

Since the value collected from a menu field is one of a distinct set of values, and cannot be misspelled, Visio Logger can use this fact to create alerts, which will notify anyone in the facility to the effect that an exception has occurred.

Value based Exception Alert

Note in Figure 7 the "Exception value" drop down list has the value "Bob Jones" selected. Only values in our Available values would be in this list.

When something is chosen from that list, then, visiting the "Exception" tab, also shown in Figure 7, displays the following options:



Who to See ? ● Data/Signature field ○ Action field

Exception Background Color 🟨

Exception Text Color ⬛

The quick brown fox jumps over the lazy dog

This alert colors the data cell when the value provided by the signer equals the Excption value. Choose the exception value on the Data Entry tab.

This alert is only active when the "choose value" data entry method is chosen.

Figure 8 Value Based Exception - partial view

In the above figure, we have selected the button to the right of “Exception Background Color” and chose yellow to represent our exception.

Now, whenever a visitor comes into the facility and selects “Bob Jones” from the menu for whom they are there to see, any computer using Visio Logger (which will see the log of visitors in real time), will see the *cell* for *this* menu item for this visitor highlight in yellow, and will know that this visitor has triggered this exception.

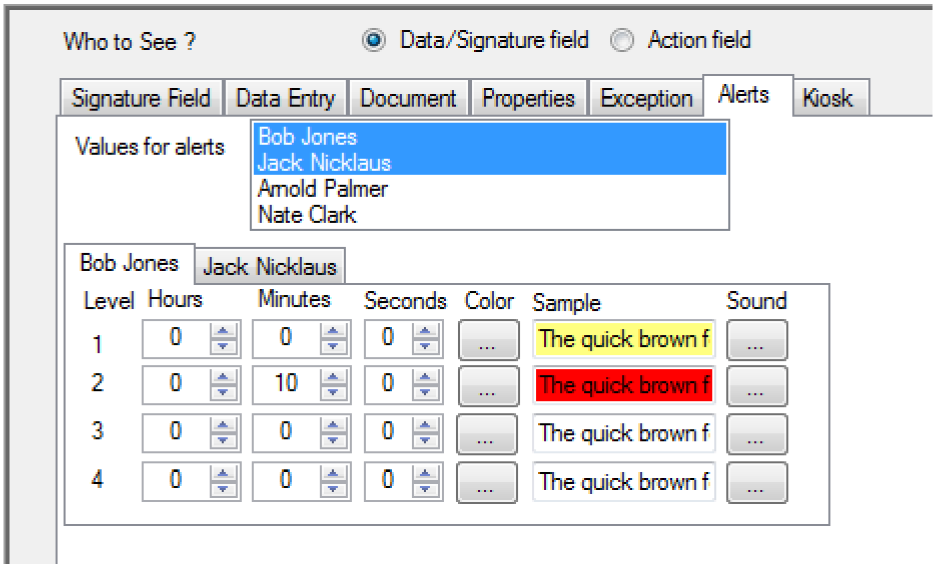
A very real example is when a facility offers certain services, but often people come in for services offered at the building next door. Providing a menu entry for the *other* building’s services would immediately alert staff to the effect that the visitor is in the wrong building rather than have the customer wait until it is their turn for service to find out.

Duration based Exception Alert

Later in the document, we describe how to create Action fields. For now, we introduce them as a simple way to track time and activity for a given visitor record, in order to have the concept of time, or more specifically, *elapsed* time, on the table to discuss the creation of Alerts.

Therefore, assuming an Action field has been created, and it is *intended* to be used at some point *after* the visitor has signed in, we can start to track the amount of elapsed time from when that visitor signed in, until any action occurs

Visiting the “Alerts” tab, which is second from the right in Figure 7, we see, again, a partial view:



Who to See ? ● Data/Signature field ○ Action field

Values for alerts: Bob Jones, Jack Nicklaus, Arnold Palmer, Nate Clark

Bob Jones | Jack Nicklaus

Level	Hours	Minutes	Seconds	Color	Sample	Sound
1	0	0	0	...	🟨 The quick brown f	...
2	0	10	0	...	🟥 The quick brown f	...
3	0	0	0	...	The quick brown f	...
4	0	0	0	...	The quick brown f	...

Figure 9 Time based Alerts – partial view

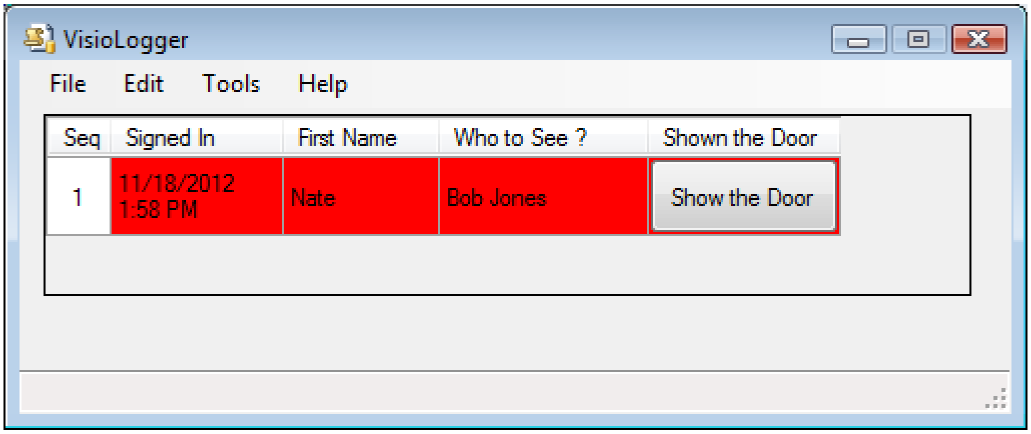
In the middle list in the above figure, you can see the available values for our menu field. Selecting any one or more of them causes the tabs to appear below the list. You can set up one of these alerts for any of the values in your menu.

We are showing what happens if a visitor chooses to see "Bob Jones". Note the table of elapsed time. This is an amount of time that may elapse between when the visitor signs in and *any* action occurs. It does not matter specifically what that action is, just that there is a time (which is another name for action, in this case) field available. Note that action fields become time stamps when the action is taken and have an empty value until then. Alerting works by continually elapsing time until at least one of these time stamps has a value.

There are 4 levels of alerts. Therefore, the row of the data can turn up to 4 different colors based on how long it's been. In our case, we have chosen that after 0 minutes and 0 seconds, the row will turn yellow. Essentially, it will be yellow from the moment the visitor signs in, which is the same (in this case) as the value based alert we previously defined.

However, if, after 10 minutes, some action did not occur on the record, that row will now turn red.

Here is our first visitor sign in row with this alert in place. Visitor Nate has been in the building for more than 10 minutes, and he has not been informed that Bob Jones is not here, in other words, he has not been "Shown the Door"

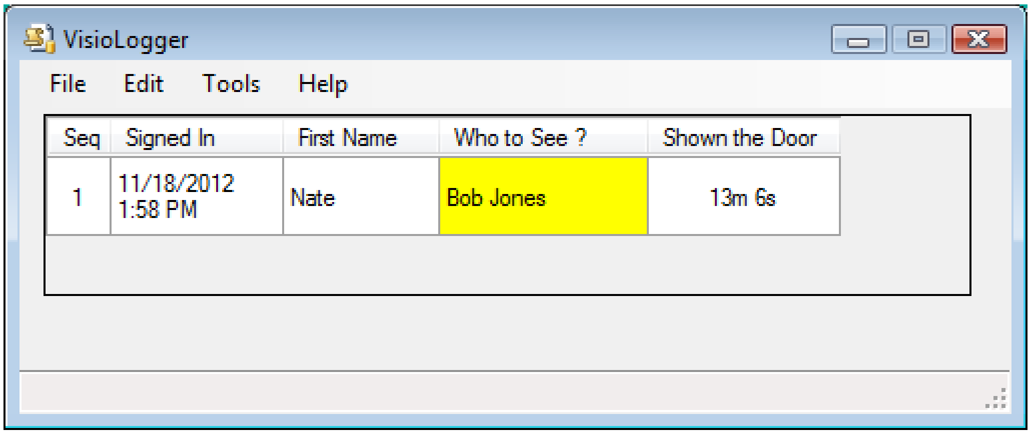


Seq	Signed In	First Name	Who to See ?	Shown the Door
1	11/18/2012 1:58 PM	Nate	Bob Jones	Show the Door

Figure 10 Our first alerting record

Once somebody sees Nate and finds he's looking for Bob Jones, who is not here, they can inform him, and click the appropriate button in Visio Logger to track the time and duration of his visit.

After clicking this button we see:



Seq	Signed In	First Name	Who to See ?	Shown the Door
1	11/18/2012 1:58 PM	Nate	Bob Jones	13m 6s

Figure 11 Time based alert dispatched

In the above, we are back to the normal Visio Logger view of the data, which, again, any computer in the facility can see (and could have pressed the button on our action).

Visitor Nate was in the building for 13 minutes and 6 seconds before someone noticed the alerts and notified him that Bob Jones is no longer here.

Though the above example is tongue in cheek, one can imagine some serious reasons for having alerts. In a hospital check-in system, for example, if the "Symptom" selection were "Chest Pain', the facility would find significant value in having these alerts setup, against an action such as "Triage started" perhaps.

Action Fields

We've talked about Action Fields in the context of using them for the alerting mechanism. Now, we discuss the details of creating them.

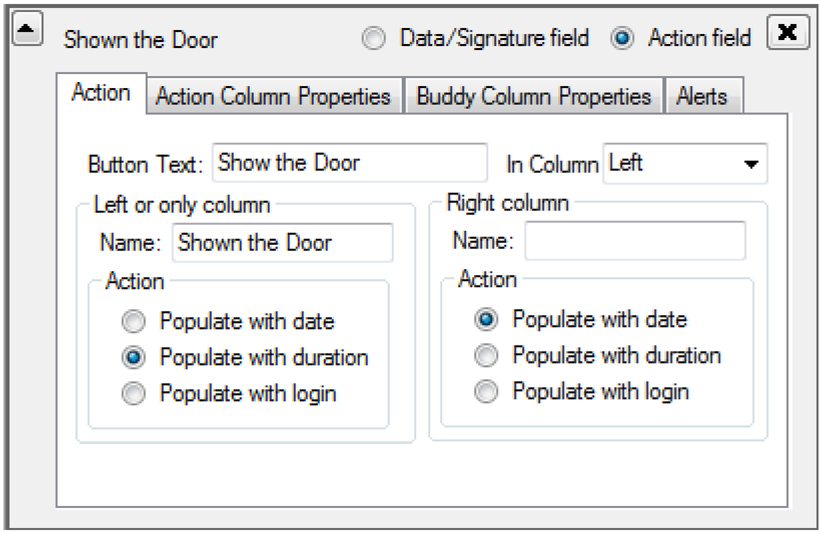
As with all fields, we press the ⊞ under the "Page #" for the "Page" we want our field to appear on.

Note, however, that in this case, Action Fields are never actually shown to the signer (or visitor), they represent a "button" in the columns of data that appears for each visitor that some *staff* in the facility would press at some future time[8].

Therefore, it does *not* matter which page the action field is on. Further, even though a page that contains a data entry field that uses the keyboard or pad menu can only have *one* such field, it can still have any number of Action Fields.

The only reason to consider what page to put an action field on, is where do you want it to appear, left to right, in your columns of data that you are collecting. Specifically, all of the fields on Page 1 are the left most columns in your data. And their order is based on how they are *vertically* ordered in that Page's fields definitions (you can move a field up or down on a single page). From top to bottom, the first field on a page is the leftmost, the last one, the rightmost, and so on.

We have pressed the ⊞ button and a new field was placed below any existing field(s) on the page. The first thing we do is select the "Action field" radio button at the top of the field definition pane, and we see:


Shown the Door ○ Data/Signature field ● Action field

Action | Action Column Properties | Buddy Column Properties | Alerts

Button Text: Show the Door In Column: Left

Left or only column
Name: Shown the Door
Action
○ Populate with date
● Populate with duration
○ Populate with login

Right column
Name:
Action
● Populate with date
○ Populate with duration
○ Populate with login

Figure 12 Creating an Action Field

It is important to click the Action field radio button in order to see the settings as shown in Figure 12.

[8] However, there are automated ways for one of these action(s) to occur. For example, when "Sign Out" is one of the desired activities (you are implementing *both* sign in and out), you can configure one of these actions to mark the time of sign out and the button will be pressed "by Visio Logger" when the visitor signs out.

In this case, the Name of the column is specified in the Name: field within the “Left or only column” group. The row of data will have a button in this column, and we indicate the text on that button in the “Button Text:” field.

Usually, you can consider the name of the column as the action taken, and the text on the button as the verb to take that action.

Also, note, in the “Left or only column” group, are 3 “Action” options, Populate with date (or duration), or with login.

Populate with date means to timestamp the cell in that row with the exact time that the action occurred. Populate with duration means to mark the amount of time that has elapsed since the visitor first started signing in. Note that the “Signed In” column is built into every Visio Logger Database and is automatically tracked for every visitor. It should also be remembered that the date and duration selections are interchangeable. For example, if, in the future, you would rather look at duration instead of date, you can switch the Action to duration, and the existing data in existing rows will have the necessary information to display duration instead of date.

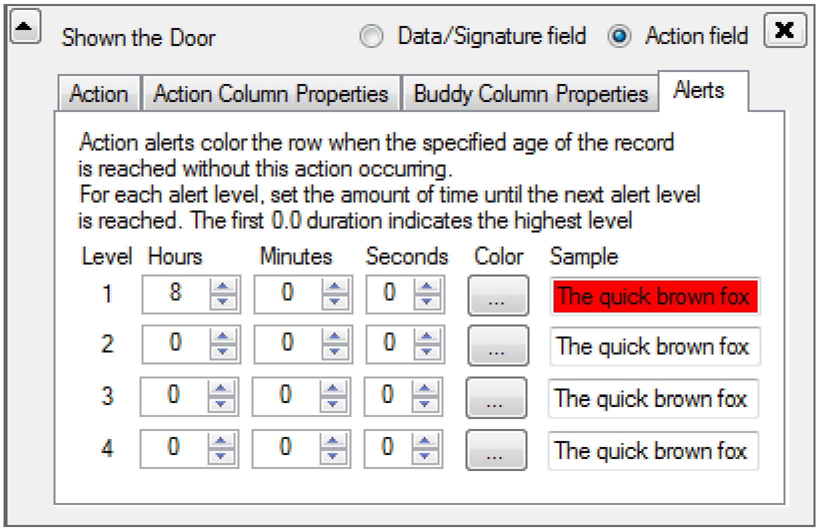
The “login” option is a way to indicate *where* in the facility the button was pressed. You could, for example, have people signing *out* in a different place then signing in. Or, you could have different stations around the facility performing different services. Essentially, the value in the cell upon pressing this button will tell you the login name of the person signed into the computer, which should lead to understanding where in the facility the action was taken for the visitor.

Buddy Columns

Figure 12 shows a group of options for a “Right column”. Essentially, this allows you to track *both* time *and* location. You can choose either date or duration for the left column, and then login for the right column (or vice-versa), and this will then provide more information about the action. Remember to provide a name for the Right (Buddy) column so that it will appear within your rows.

Action Alerts

These types of alerts are much like the duration based alerts that are triggered from certain values in one or more *menu* fields. However, they are based simply on the age of the record, not on the value of any one field in the record.



▲ Shown the Door ○ Data/Signature field ● Action field ☒

Action | Action Column Properties | Buddy Column Properties | Alerts

Action alerts color the row when the specified age of the record
is reached without this action occurring.
For each alert level, set the amount of time until the next alert level
is reached. The first 0.0 duration indicates the highest level

Level	Hours	Minutes	Seconds	Color	Sample
1	8	0	0	...	🟥 The quick brown fox
2	0	0	0	...	The quick brown fox
3	0	0	0	...	The quick brown fox
4	0	0	0	...	The quick brown fox

Figure 13 Duration Based Action Alerts

Here we are more concerned with something not happening within a period of time, for example, someone not signing out after having been in the building for a certain number of hours.

In Figure 13, we see a definition where the entire row of data will turn red after the visitor is not "Shown the Door" more than 8 hours after signing in. If you have a Sign Out action, and have configured Visio Logger to perform Sign in *and* out, such an alert will immediately tell you who has not signed out, in the current day, or for the entire history of the data you have collected so far.

Other items in Figure 13 include:

- Action and Buddy Column Properties
  These are the background and text colors for the cell before and after the cell is provided a value. Note that the button does not always fill the cell when it is visible, so the color behind the button can be seen. Once the button is pressed, the cell is considered "changed", and its background and text color *can* be different than the default of white with black text.
  These kinds of properties are available for Data Entry fields as well. The primary use for them would be to change the color of an "Editable" text field, so that the fact that a field was edited would be known because that field would then be in a different color.
- The field can be moved up or down in the vertical order.
  The up arrow in the upper left of the figure shows that this field can be moved "up" in the order, which would move it "left" in the columns display. If there were a field currently below this field, there would also be a "down" arrow to move the field down.
- The field can be deleted by pressing the ☒button in the upper right.

Handwriting or Signature Fields

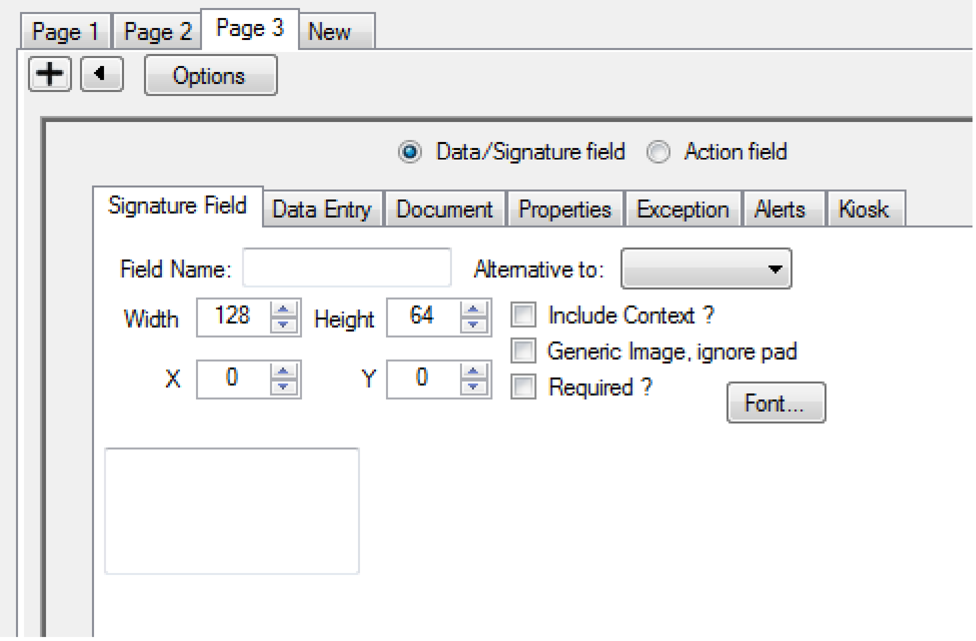
In Visio Logger, handwriting fields are represented as bitmaps on each row of data for the visitors.

These bitmaps are created from the series of points generated when a user writes on the signature pad with the stylus.

We can start with Figure 4, which we arrived at by first choosing Edit → Columns from the Visio Logger main menu[9], and then pressing the + button underneath the tab for the Page where our signature field will be (or we can select the "New" tab and create a new page).

Steps to create a handwriting[10] field are:

1. Switch the active tab from the "Data Entry" tab, to the "Signature Field" tab. Ensure that the properties you see are as in the below figure.



● Data/Signature field ○ Action field

Field Name:

Alternative to:

Width 128 Height 64

X 0 Y 0

□ Include Context ?

□ Generic Image, ignore pad

□ Required ?

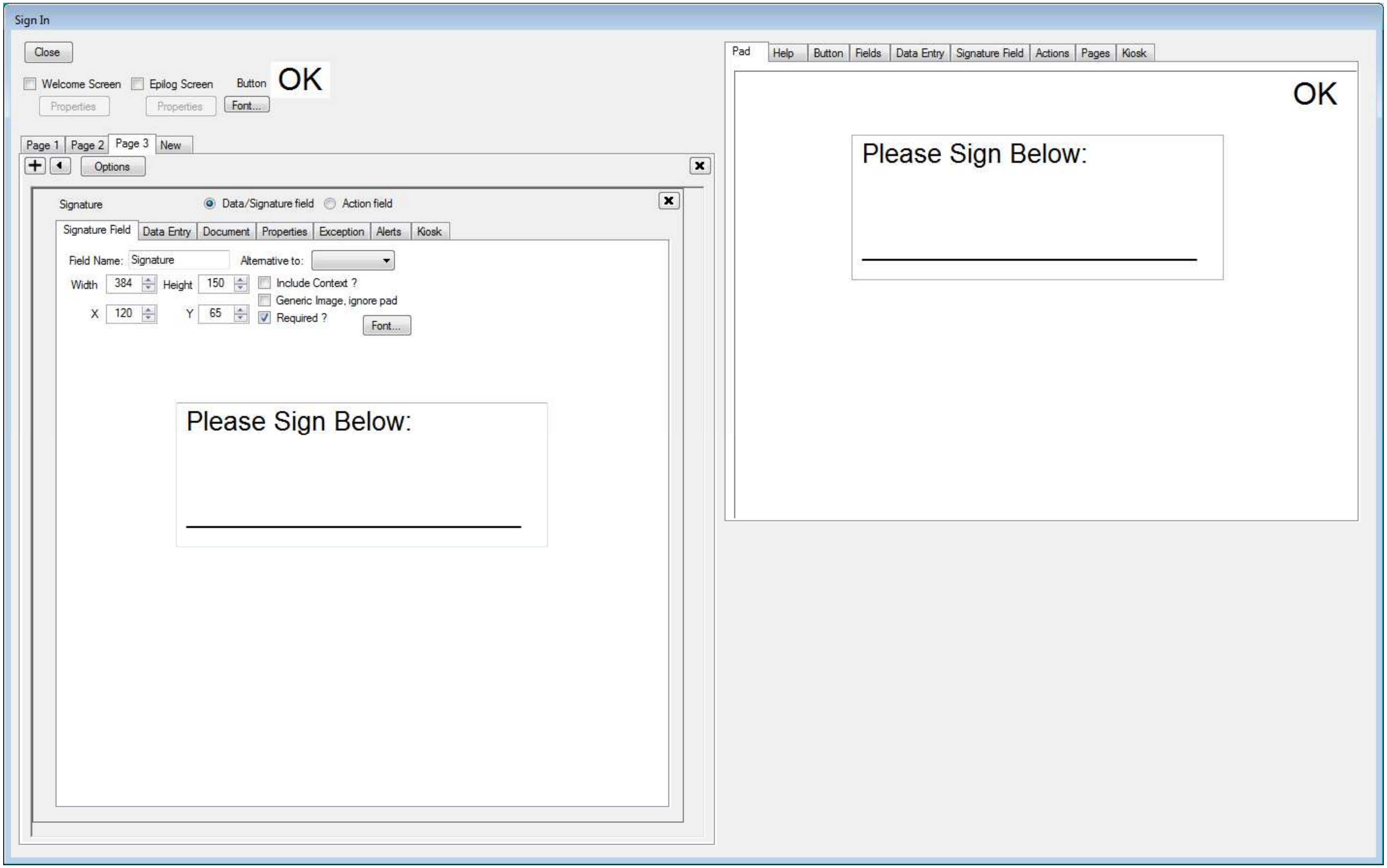
Figure 14 Creating our first Signature Field - partial view

2. Fill in the "Field Name:" value

3. Ensure that "Generic Image, ignore pad" is **not** selected

[9] Remembering to run "As Administrator"

[10] We use both terms handwriting and signature to stress the point that it does not have to be a signature



□ Welcome Screen □ Epilog Screen Button OK

Signature ● Data/Signature field ○ Action field

Field Name: Signature Alternative to:

Width 384 Height 150 □ Include Context ?

□ Generic Image, ignore pad

X 120 Y 65 ☑ Required ?

Please Sign Below:

Figure 15 Our first Signature Field

Referring to Figure 15 we performed the additional steps:

4. Defined the Font for the text displayed on the signature pad. In the white area in the left of the figure (near the top), use the Font button to select the font. We chose Arial 20 pt.

5. Specify the width and height of the signature capture box, we chose 384 and 150 respectively.

6. Type the desired prompt into the signature box. Note carefully, this is the outlined area that is in the lower left of Figure 14. It will change in size with the width and height adjustment. We typed "Please Sign Below:", pressed the [Enter] key a few times, then drew a line by pressing the "_" key multiple times. Based on your selected font, you may need to adjust the number of times you press [Enter], or adjust the height of the capture area.

   Note that, when either a background image or text is specified, the *text* specified in this prompt is *not* sent to the pad. Please refer to Figure 3 for more information.

7. Now, in the *right* side of Figure 15, we click *in* the signature capture area and, holding the mouse button down, *drag* it to where we want it to appear on the signature pad. Note, only handwriting that falls within this area will be captured, all other handwriting is "clipped".

8. We selected to Font for the OK button. In the upper left of Figure 15 is a button to set the font for the OK button that the user will tap on the pad. Note that we could ALSO set the Text of the button, to "Yes", for example, by clicking it (just above the Font button) and entering the different value.

9. Finally, we "Clicked and dragged" the Ok button from its default location (upper left), to the upper right, again, in the right side of Figure 15.

Note that any one page can have any number of handwriting fields on it. Because of this, we could display an entire form on the page using multiple fields and take as our input locations just the handwriting locations and split those into several columns in our data.

This defines our Signature Field. Note that, as with all pages, you can change the order that the signature is collected by sliding the entire page to the left or right.

Some other aspects of Figure 15 not described before are:

- Include Context ? This option includes the "background" from the signature in the bitmap of the signature that is created. Essentially, in the current case, the stored signature will also contain the text "Please Sign Below:" and the underline exactly as seen in the figure, with the addition of the handwriting.
- Generic Image, ignore pad. This option allows you to define generic bitmaps for your database. When selected, you can create a bitmap of any size (not limited to the size of your signature

pad) and this bitmap can be supplied using the Visio Logger integration techniques.

Note that Visio Logger ships with one such integration tool designed specifically to use one or more of these generic bitmaps. Did you know that you can take a picture of your signer, at any point during the sign in process, or later via the use of an Action field? For more information, please reference InnoVisioNate Video Tool on page 60.

- The “Required ?” option allows the system to continue on to the next page, even if the signer simply taps the Ok button on the pad without writing anything.
- The “Alternative to:” option was referred to in the description of pad Keyboard entry fields. This option allows you to use *this* handwriting field as an *alternative* to some pad Keyboard entry field. If, for example, your signers may not be familiar with a keyboard, and they prefer to write the information (name, for example), then this particular option relates this handwriting to that data entry field.

That is all there is to creating a signature field.

Keep in mind also, that if you are configuring Visio Logger as a Sign In *and* out system, you can capture a signature for both activities and keep them separate. This allows you to visually inspect whether it was truly the same person signing out as in.

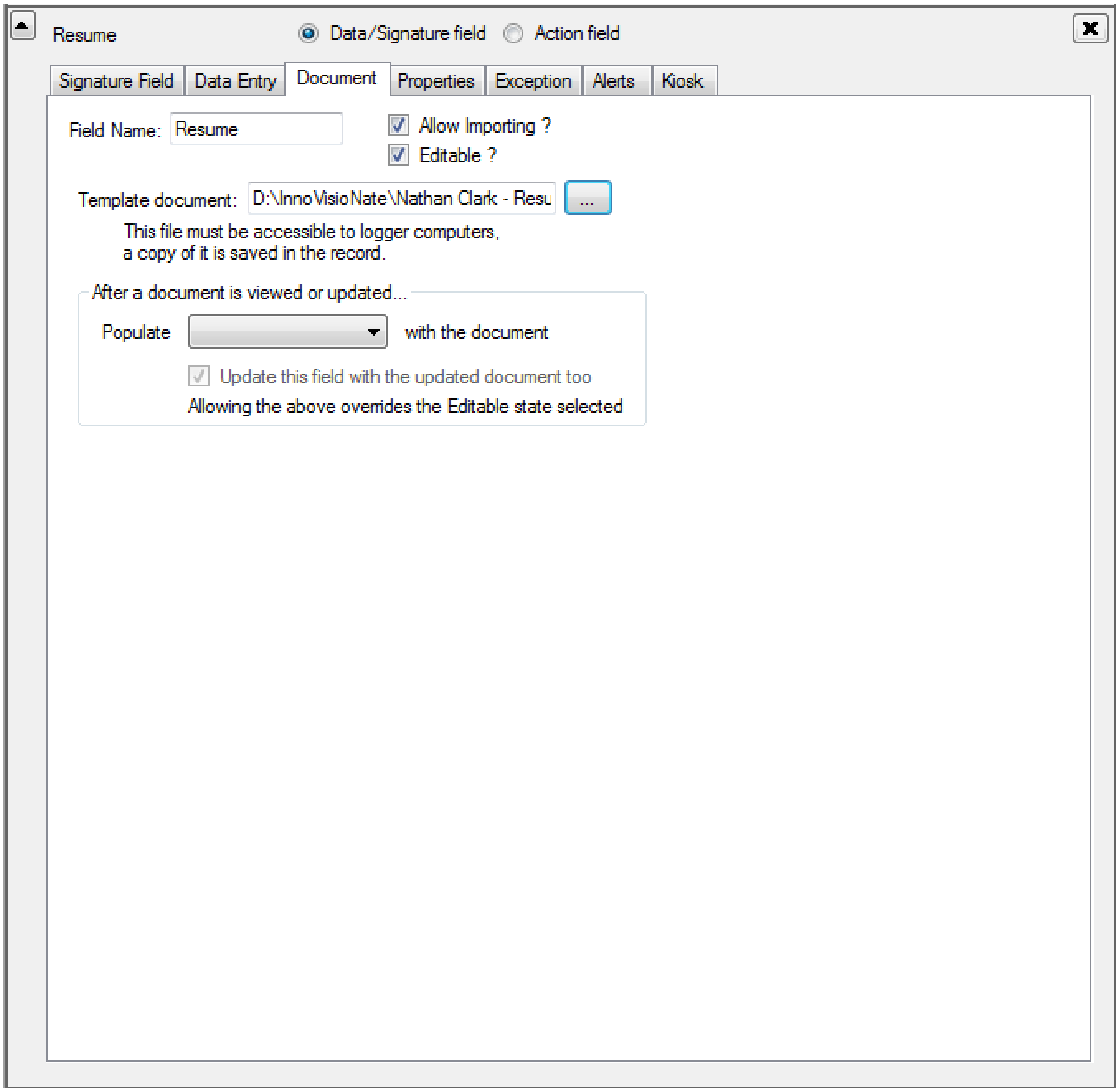
Document Field

Visio Logger can store a copy of any document for *each* visitor and staff using Viewer Mode can retrieve a copy of that document at any point by clicking the link to it stored in the cell.

It is important to remember that this document is a *copy* of a template document that is stored in some location visible to the Visio Logger computer.

Because of this, the document is essentially the same for each visitor to start out with. However, Visio Logger *can* launch the document’s application (in most cases), and it (Visio Logger) will wait for the launching application to close, and place the updated copy back in the database.

To create a Document field, we once again press the ⊞ button under the “Page #” for the “Page” we want our field to appear on. In this case, the selection of the Page # is made solely on where (left to right), you want the document cell to be in the data view. Additionally, you can put any number of Document fields on any page, including pages that contain a pad Keyboard field or a pad menu field.



Field Name: Resume

☑ Allow Importing ?

☑ Editable ?

Template document: D:\InnoVisioNate\Nathan Clark - Resu

This file must be accessible to logger computers,
a copy of it is saved in the record.

After a document is viewed or updated...

Populate [ ] with the document

☑ Update this field with the updated document too

Allowing the above overrides the Editable state selected

Figure 16 Creating a Document Field

Note in Figure 16 that we navigated to the Document Tab, entered a name for Field Name, and specified a template document, which will be initial copy of the document in the cell.

Other options include:

- Allow Importing? Visio Logger will place a link in the data cell where a user could replace the document in the cell by importing a new one.

- Editable? This option provides a link to the document in the cell that will launch the associated application, and wait for that application to finish before putting the document back in the cell.

- Populate another Document Field with the document. This feature places the document from *this* field into a *different* field if a user edits the document from this field.

  You can use this powerful feature to "migrate" a document down the chain, for example, to have multiple staff successively edit the document. As a document is copied into a different field, the date and time that was done is displayed in that different field. Thus, staff would know when it is their "turn" to add their changes to it, after which, it automatically flows down the line.

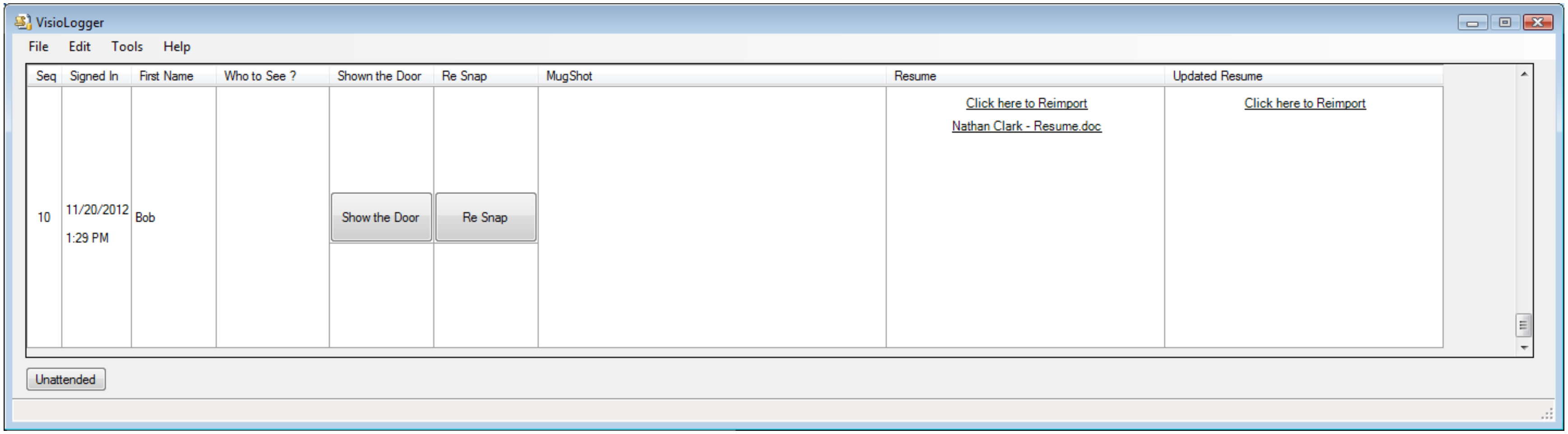
We will provide an example of the "migration" feature.

In Figure 16, we've created a Resume Field. As each visitor signs in, this field will contain a copy of a word document from the local machine or network.

We also created an additional Document field, and called it "Updated Resume". But, in *this* case, we did not specify a template document. The template document field is not required, and if not provided, the cell will contain an "Import file" link (if the "Importable ?" option is also selected).

Now, back in Figure 16, the "Populate another Document Field" list *will* have "Updated Resume" available. So, we choose that item in the list, then click Close to finish editing the columns.

Running Visio Logger in Logger mode, we sign in a visitor, which we show with the following figures.

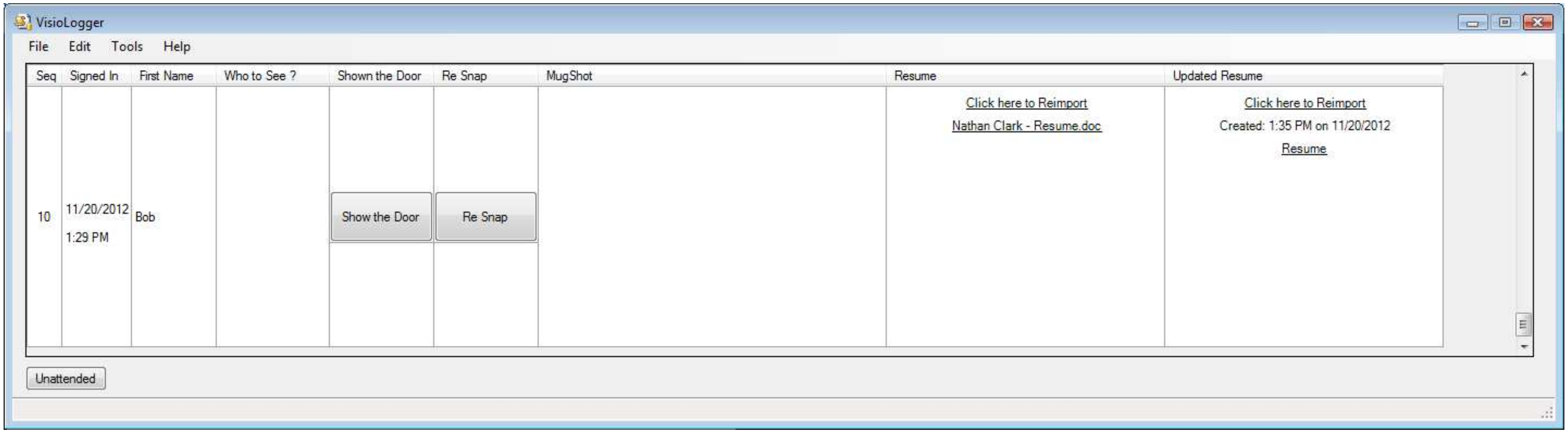


Seq	Signed In	First Name	Who to See ?	Shown the Door	Re Snap	MugShot	Resume	Updated Resume
10	11/20/2012 1:29 PM	Bob		Show the Door	Re Snap		Click here to Reimport Nathan Clark - Resume.doc	Click here to Reimport

Figure 17 Visitor Bob has just entered his name

When the row is first created, Document cells with a template document specified contain a copy of that document. Note the Resume cell allows re-import. And, it also provides a link to edit the document. The "Updated Resume" field is empty.

Any computer in the facility running Visio Logger could now edit this document. Clicking on the file name field in the Resume cell launches Word, and Visio Logger will wait until Word closes before putting that file back in the database. Assuming we have done this, note that the document is *now* in the "Updated Resume" document field, as well as information about when, and from what *field* the document came from!



Seq	Signed In	First Name	Who to See ?	Shown the Door	Re Snap	MugShot	Resume	Updated Resume
10	11/20/2012 1:29 PM	Bob		Show the Door	Re Snap		Click here to Reimport Nathan Clark - Resume.doc	Click here to Reimport Created: 1:35 PM on 11/20/2012 Resume

Figure 18 After editing the document in Resume

Note carefully that the "source" field ("Resume") in the information shown for the "Updated Resume" field above, is displayed. This is because multiple document fields could be targeting the "Updated Resume" field and this tells you where this document came *from.*

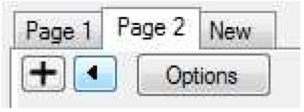
Document Fields are a very powerful way to create an information rich *system* with Visio Logger. There is no limit to the number of fields you can define which provides the opportunity to create a full document management system with Visio Logger – Out of the box.

Page Submission Order

Now that we have more than one Page of data entry fields defined, we can discuss the *order* in which fields are presented to the signer.

The "order" the pages are presented is from 1 thru n, where n is the total number of pages defined[11].

Therefore, to change the order pages are presented, you change their page #. All you need to do is "slide" a page left or right to decrease (or increase) it's page number.



For example, referencing the above image, the blue "left" button is highlighted. Pressing that button will case the current page (Page 2) to slide to the left – swapping places with the current Page 1. If there were 3 or more pages, Page 2 would also have a button to slide it to the right.

Using our 3 currently defined fields, you can now ask the visitor who they are here to see, what their first name is, and to provide their signature. And, you can do this in any order.

Deleting pages

Any page can be deleted by pressing the "X" button in the upper right of the page.

Please note that all fields on the page will also be deleted, and that the action is not undoable.

When all pages and fields have been defined, press Close in the upper left of Figure 4 and Visio Logger will return to Viewer Mode. You are now ready to start collecting data, select Edit → Station Type → Logger from the main menu to try out your new visitor management system.

Configuring Auto Logger operation

Auto Logger is a special operational mode within Visio Logger that is used to turn your system into both a Sign IN *and* Sign Out visitor logging tool.

Now that we have Pages and Fields defined for collection, we are in a position to configure the system to use Auto Logger.

[11] Not all pages are required to have any data fields on them, there may be pages with just action fields on them, or with fields that are "Not on the Entry Panel". Such pages are skipped.

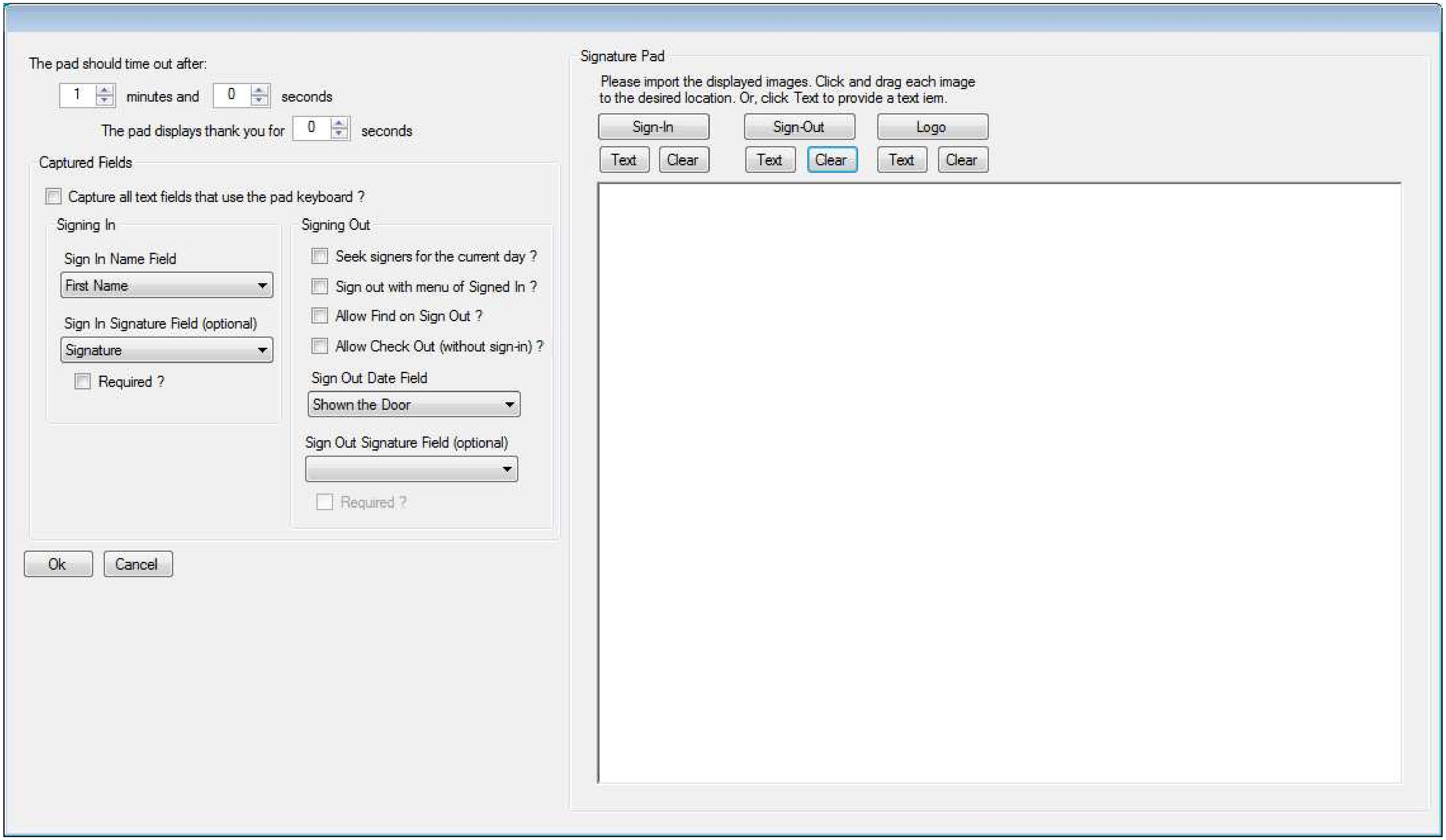
It is important to remember that Auto Logger mode is used for less complex data needs, especially where “flow” through the data is not necessary. The term “flow” refers to the fact that Visio Logger *can* present certain fields to the visitor for entry, or **not**, depending on answers to previous questions. In other words, multiple “types” of visitors may be arriving, and certain fields NOT pertinent to one of these types can be “filtered” out to avoid any confusion amongst your visitors.

Essentially, Visio Logger in Auto Logger mode prepends the sign in process with a “question” of whether the visitor is signing in or out. If the answer is “Out” – then Visio Logger presents one of the fields the visitor signed in with in order to be able to “find” that visitor’s sign in record with the value that they, upon signing out, enter into that field.

Configuring Auto Logger mode is a matter of telling the system which fields to use for these purposes.

If Auto Logger mode is not configured, the first time you attempt to run Visio Logger this way, it will display a dialog for you to do so. To run in Auto Logger mode, choose Edit → Station Type → Auto Logger from the main menu. You will see the configuration options shown in Figure 19.

Note that we have already filled in many of the settings to aid in the discussion.



The pad should time out after:

1 minutes and 0 seconds

The pad displays thank you for 0 seconds

Captured Fields

□ Capture all text fields that use the pad keyboard ?

Signing In

Sign In Name Field: First Name

Sign In Signature Field (optional): Signature

□ Required ?

Signing Out

□ Seek signers for the current day ?

□ Sign out with menu of Signed In ?

□ Allow Find on Sign Out ?

□ Allow Check Out (without sign-in) ?

Sign Out Date Field: Shown the Door

Sign Out Signature Field (optional):

□ Required ?

Signature Pad

Please import the displayed images. Click and drag each image to the desired location. Or, click Text to provide a text iem.

Figure 19 Configuring Auto Logger Mode

From top to bottom, left to right, we describe the Auto Logger configuration options.

First, when the station is totally unmanned, after hours, for example, it is necessary to have the system "cycle" to the beginning, in case someone starts a sign in our out, and walks away before finishing it. The "Pad time out" values in the upper left are used for this purpose. Provide values that are realistic. If a person needs, for example, a minute to remember how to spell their name, if the timeout is set at 1 minute, the pad might suddenly cycle while they're thinking of how to spell it.

Nevertheless, we've used 1 minute for this value.

Whenever a user successfully signs out, the system can display a "Thank you" message for a configurable amount of time. Leave this setting at 0 to disable this option altogether.

In the Captured Fields group, we are specifying whether *all* fields defined that use the signature pad keyboard, or a menu option should be filled in by the visitor. You could, for example, collect just that *one* field that you later need to sign that person out (described later), *or* you could have these visitors fill in all the fields they would if the system is running in normal Logger mode. It is important to remember, however, that in Auto Logger mode, the system does not utilize the integration technique that can affect the "flow" of data through your fields. In other words, if this option is on, your visitor will have to enter data for ALL fields that you've defined, which does not have to be the case in Logger mode.

Sign In group options

Next, when Signing IN – which field is THAT field which, upon Signing OUT – will be presented to the signer in order for the system to FIND that sign in record?

Also, on Signing In – do you require a signature, and if so, which of the available signature fields will be used to store this signature. Auto Logger mode is not able to use more than one handwriting field on Sign In. Though, it *is* able to use separate handwriting fields for Sign In and Sign Out.

Sign Out group options

When the visitor chooses to sign out, the properties that you specify in this group change the sign out experience in certain ways.

Recall that Sign Out essentially just presents the visitor with the one field that you indicated was the "Sign In Name Field" just above.

However, after or while asking the visitor for this information, these options come into play.

- Seek signers for the current day. When Visio Logger is trying to find the signed in users (during sign out), it only looks at rows of data that are currently in "view" to find the visitor's record. This option is only effective when the current "view" includes visitors from any *prior* day (see Edit → Preferences to see how to change the current view). You must set this option to ON if you do NOT want Visio Logger to consider visitors who signed in on a prior day AND when more than one day of data is showing in the "view" of data.

- Sign out with menu of Signed In. This option instructs the system to present a menu of currently signed in visitors (including their Name value, and the time they signed in), so that the visitor simply has to find their name and sign in time on that menu and tap it with the pad stylus.
- Allow Find on Sign Out. With this option, the user can begin tapping in their name on the pad keyboard, and tap a Find Key, also on the keyboard, to tell the system to create a menu of signed in user's whose name starts with what they have already entered.
- Allow Check Out (without sign-in). If a visitor never signed in, perhaps they entered the facility during a time when sign in wasn't required, but leave the building when they'd otherwise have to sign out, then this option provides a special "Tap here if you didn't sign in" key on the keyboard and the system will create their sign out record without actually matching it with any sign-in record.
- "Sign Out Date Field". At least one field in your defined fields must be an "Action" field. You select that field in this drop down list and Visio Logger will use it to timestamp the data with the time that the person signed out.
- Sign Out Signature Field. You can define an additional signature field that contains the visitor's signature upon sign out. If this value is set, the system will gather this signature after the visitor enters the Name field value, and place it in the row of data for that visitor.

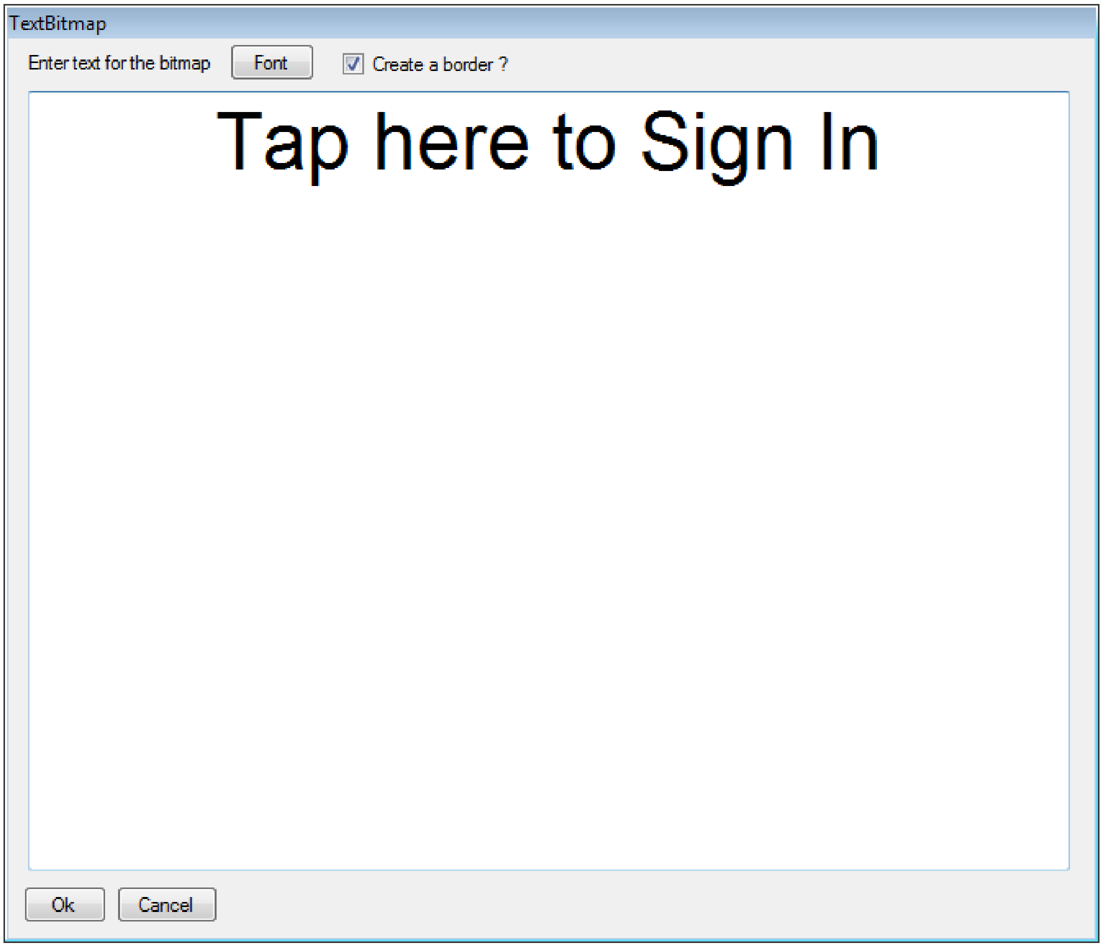
The Sign In/Out signature pad screen

The area to the right in Figure 19 shows the configuration of the initial, or "resting" image on the signature pad while the system is waiting for the next visitor.

With these options, we will indicate where on the pad the visitor will tap to start either sign in, or sign out.

You can use images for this purpose, but for the sake of simplicity we will create simple text areas to accomplish this.

First, click the "Text" button underneath the "Sign In" button (f you were going to provide an image, you would click the Sign In button itself). The following figure shows the text we have entered for our prompt. Before entering this text, we clicked the Font button and set the font to Arial 36 pt (which is ½ inch high on the pad), and, we selected the "Create a border ?" option.

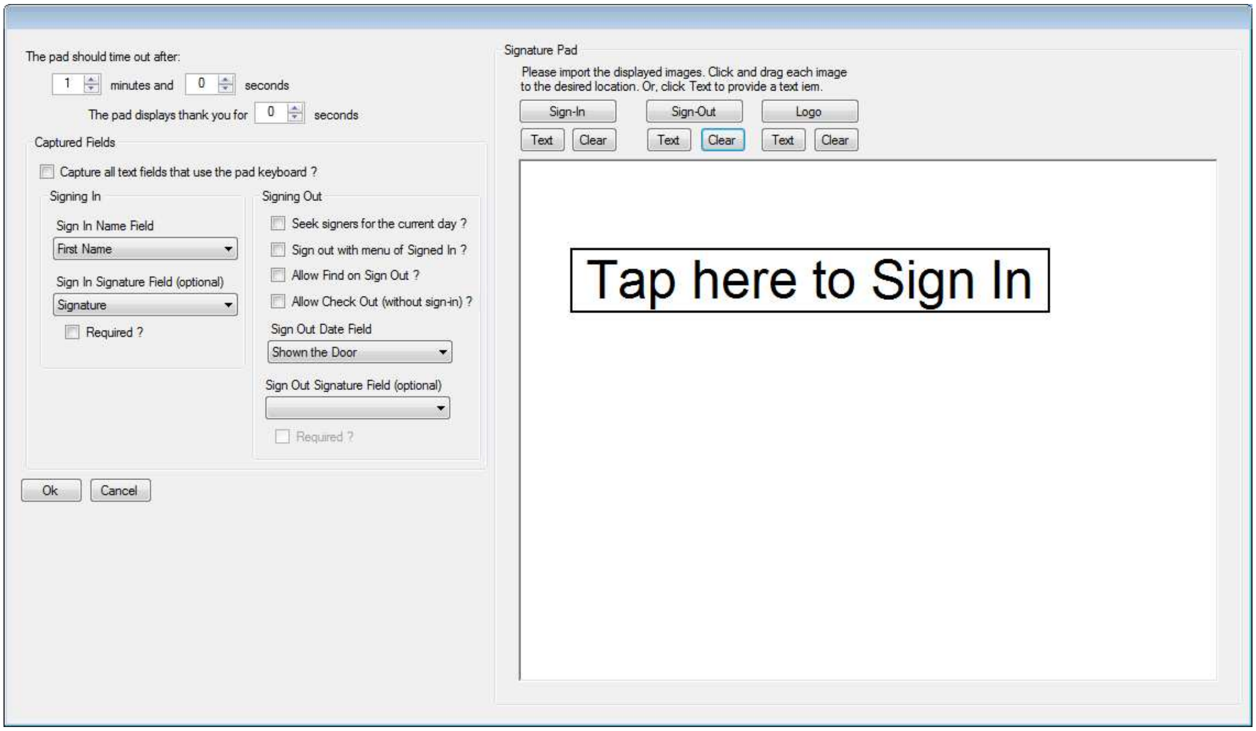


☑ Create a border ?

Tap here to Sign In

Figure 20 Creating a Sign In area for the pad

Upon pressing Ok in Figure 20 we see the following, back at the Auto Logger configuration window.



The pad should time out after:

1 minutes and 0 seconds

The pad displays thank you for 0 seconds

Captured Fields

□ Capture all text fields that use the pad keyboard ?

Signing In

Sign In Name Field: First Name

Sign In Signature Field (optional): Signature

□ Required ?

Signing Out

□ Seek signers for the current day ?

□ Sign out with menu of Signed In ?

□ Allow Find on Sign Out ?

□ Allow Check Out (without sign-in) ?

Sign Out Date Field: Shown the Door

Sign Out Signature Field (optional):

□ Required ?

Signature Pad

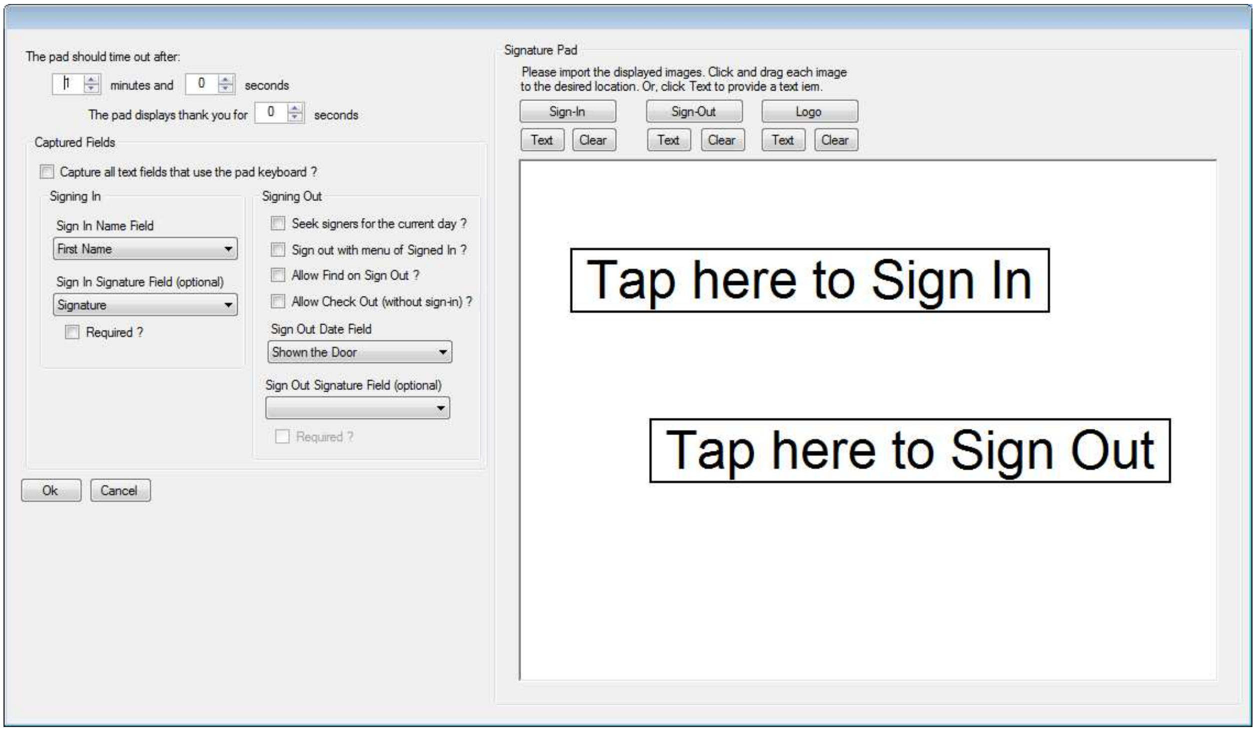
Please import the displayed images. Click and drag each image to the desired location. Or, click Text to provide a text iem.

Tap here to Sign In

Figure 21 Positioning the Sign In prompt

Note also that we clicked and dragged the Sign In prompt to the location shown. Otherwise, it would remain in the upper left corner of the window.

Next, we click the "Text" button directly beneath the Sign Out button, provide similar information, dispatch that dialog, and finally, after moving *that* prompt we see our final configuration shown in Figure 22



The pad should time out after:

1 minutes and 0 seconds

The pad displays thank you for 0 seconds

Captured Fields

□ Capture all text fields that use the pad keyboard ?

Signing In

Sign In Name Field: First Name

Sign In Signature Field (optional): Signature

□ Required ?

Signing Out

□ Seek signers for the current day ?

□ Sign out with menu of Signed In ?

□ Allow Find on Sign Out ?

□ Allow Check Out (without sign-in) ?

Sign Out Date Field: Shown the Door

Sign Out Signature Field (optional):

□ Required ?

Signature Pad

Please import the displayed images. Click and drag each image to the desired location. Or, click Text to provide a text iem.

Figure 22 Final Auto Logger configuration

This completes the configuration of Auto Logger mode, pressing Ok in the above figure will now place the system into this mode, and no further configuration is required. Visio Logger can now automatically collect Sign In AND Sign Out data from your visitors for as long as your field(s) definition do not change in any way that will affect these configuration settings. If you do need to reconfigure this mode, there is a “Configure” button in the bottom right of the Signature Pad display window showing while in Auto Logger mode.

Other Configurable Items in Visio Logger

We have focused on pages and fields so far in our discussion. However, there are other higher level aspects of the data that are important and we cover those topics here.

Visio Logger Processes

A “Process” is a high level entity that simply forms a container around Visio Logger Pages and Fields as described above.

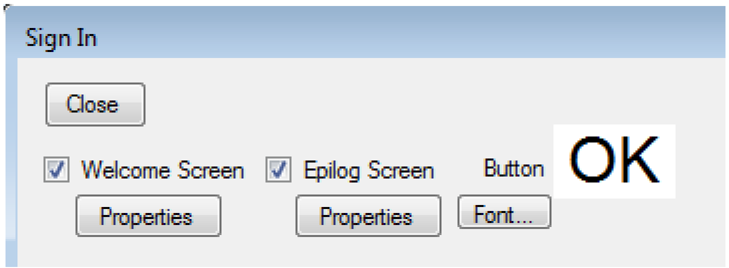
Because you can create processes, you can use the same Visio Logger database with different sets of data for multiple types of facilities, or even different types of sign in activity in the same facility. Staff using Visio Logger to view and act upon the data can also choose among these different sets of input data with a simple menu selection.

The options available at the process level are set in two different places within Visio Logger. First, we discuss those that have to do with the visitor sign in activities.

Process options related to the sign in experience

To set the options related to your particular visitor *user experience*, select Edit → Columns from the main menu, at which time you will see the window whose top portion is shown in Figure 23, this is the same window we've been using for Pages and Fields definitions throughout.

Note the process level options are whether to display the Welcome Screen and Epilog Screen, the Button field (currently OK), and the properties of the two screen types.



☑ Welcome Screen ☑ Epilog Screen Button OK

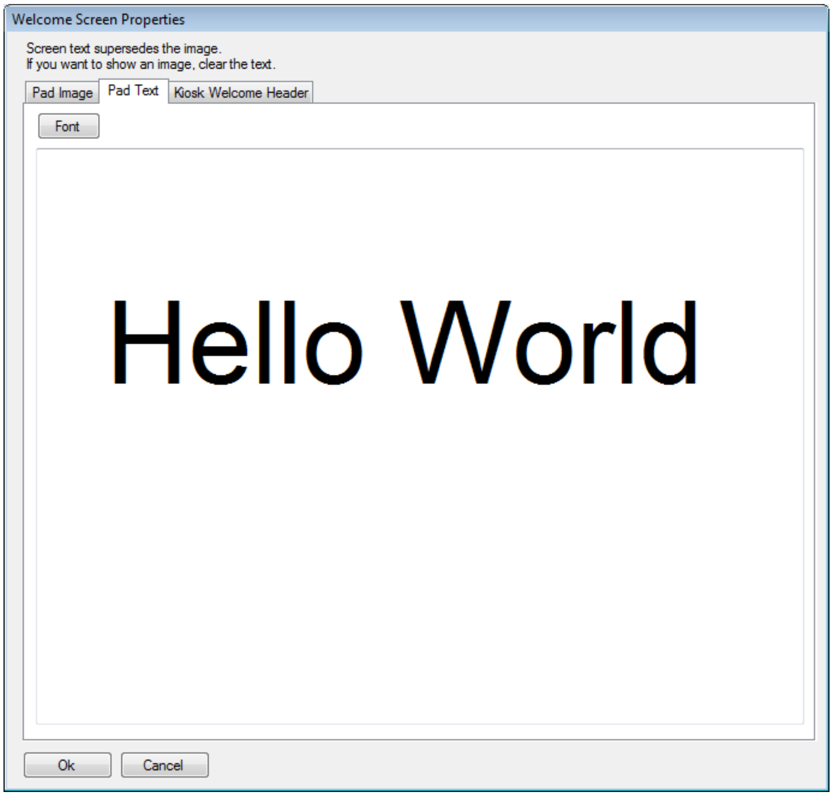
Figure 23 Process level options – partial view

Welcome Screen

The welcome screen is a page of information that is displayed on the signature pad while the pad is waiting between signers (the system is "cycled").

When a welcome screen is used, the signer must tap the signature pad with the stylus in order to continue. Therefore, typically the welcome screen would contain information to this effect.

To set the properties of the welcome screen, click the Properties button just below the Welcome Screen option.



Screen text supersedes the image.
If you want to show an image, clear the text.

Hello World

Figure 24 Welcome Screen Properties

You can use the 3 tabs shown in Figure 24 to specify what displays on the signature pad while it is waiting for the next signer. Import an image to have the pad display that image. Note that this image should be the same dimensions as the pad (if smaller, it is centered). Also note that the SigGEM Color57 pad can brilliantly display 640x480 pixel color images.

If you want to display a message or other information to the signer, use the Pad Text tab (shown in the figure) and type that text directly into the window presented. You should change the font to ensure the text is big enough for the signer to read.

The presence of any text on the Pad Text tab supersedes any image specified. In other words, the image will not show if there is any text. This allows you to create and import the image once, while easily providing text for the special days when you want to make an announcement or notification to the signers. Remembering that Visio Logger is a database application, administrative staff can specify this text, or these images, from any computer running Visio Logger and which can see the appropriate database.

None of these configuration steps necessarily need be done on the computer that is also used as the data entry station.

On the Kiosk Welcome Header page – you can also specify a bitmap that displays (during cycle time) on the upper half of the Kiosk display when the system is running in Kiosk mode. Use this image to display information to the signer on how to begin (hit the space bar to sign in, for example) or for special announcements, etc. In general, the Kiosk background bitmap combined with the Kiosk Welcome Header will be showing on your Kiosk monitor for the majority of time – when no individual is using the system to sign in.

For more information on Kiosk mode, see the document: “Configuring Visio Logger as a Full Featured Kiosk.pdf” either in the Visio Logger ProgramData directory, or on the InnoVisioNate website.

Epilog Screen

The epilog screen acts much the same as the Welcome screen and its options are specified the same way.

The difference is that the Epilog screen shows at the *end* of the sign in process, after the signer has visited every page and dispatched the last one, and it has a configurable amount of time that it is displayed before Visio Logger then cycles to the beginning of the sign in process.

You would use an Epilog Screen to display information for a short period to a) indicate to the signer that they are done, and b) to provide further instructions to the signer, such as “Please be seated”.

Button

The “Button” is simply the text that the signer, when using an electronic signature pad, taps with the pad stylus to indicate they are done writing on the pad (to “dispatch” the page).

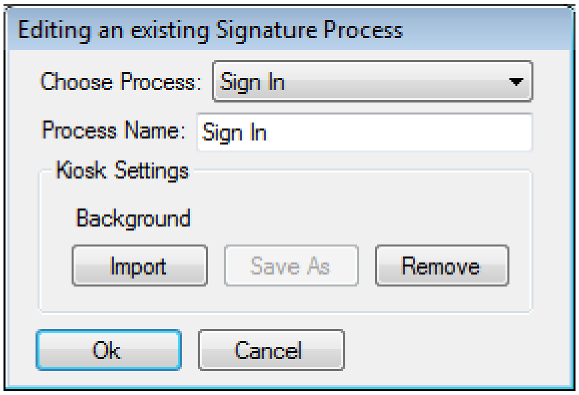
You can enter any text in this field. Typically you would adjust the font to ensure it is clear to the signer (who will see this button on the pad).

After you specify the content of the button (Yes, OK, etc), you can move where on the pad the button will display. Simply click and drag the button in the image of the pad that appears to the right of Figure 15.

Process Options related to process name and Kiosk Mode

To see these settings select File → Choose Process from the Visio Logger main menu.

Doing so will display the dialog shown in Figure 25. The point of this dialog is to choose from the existing processes in the current database. In the dialog, a list of available processes (in the current database) is shown, simply select the one you want to be the active process and press Ok. You can change the process name by entering a new name in the “Process Name:” field.



Choose Process: Sign In

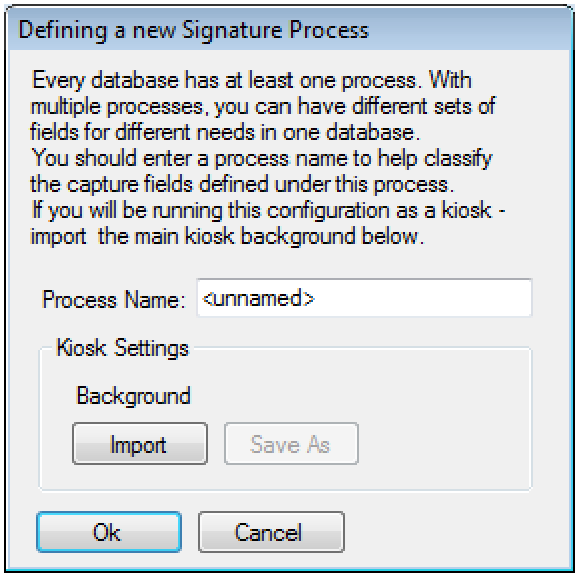
Process Name: Sign In

Figure 25 Choosing a process

Note the Kiosk Settings group of options. This is where the main background image for Kiosk mode is imported. To see more about configuring Visio Logger in kiosk mode, we refer you once again to: “Configuring Visio Logger as a Full Featured Kiosk.pdf” either in the Visio Logger ProgramData directory, or on the InnoVisioNate website.

Creating a new process

When running Visio Logger as a user with administrative privileges, choose File →New Process from the main menu. You will see a window as in Figure 26.



Every database has at least one process. With multiple processes, you can have different sets of fields for different needs in one database.
You should enter a process name to help classify the capture fields defined under this process.
If you will be running this configuration as a kiosk - import the main kiosk background below.

Process Name: <unnamed>

Figure 26 Creating a new Process

Simply enter the new name and press Ok, the Kiosk settings are as above.

Do not forget, however, that when you create a new process, there are *no* data fields defined for it. You might be surprised to see all of your data fields gone when you press Ok on the above figure. They are not really gone – it is just a reminder that data fields are connected to the currently chosen process. And, until you define the fields as described in detail in prior sections, there are none defined for this new process. This should illustrate the power of processes. You can have entirely different data definitions and sign in collection needs in one database, and you can switch between them, both for gathering data, and for viewing it, with a few mouse clicks.

The Visio Logger Database

Visio Logger has a formal and robust database structure that supports its ability to let you define the fields and data that are of interest to you. This database contains, in addition to the sign in data that you are capturing, all of the information about all of the fields that you define and maintain. All of the options for every data point (cell) that you capture are maintained in the database.

Database Providers

Visio Logger can use either a Microsoft Access database, a Microsoft SQL Server database (Server 2000 or higher), Oracle database, or the Open Source MySQL database system to store this data.

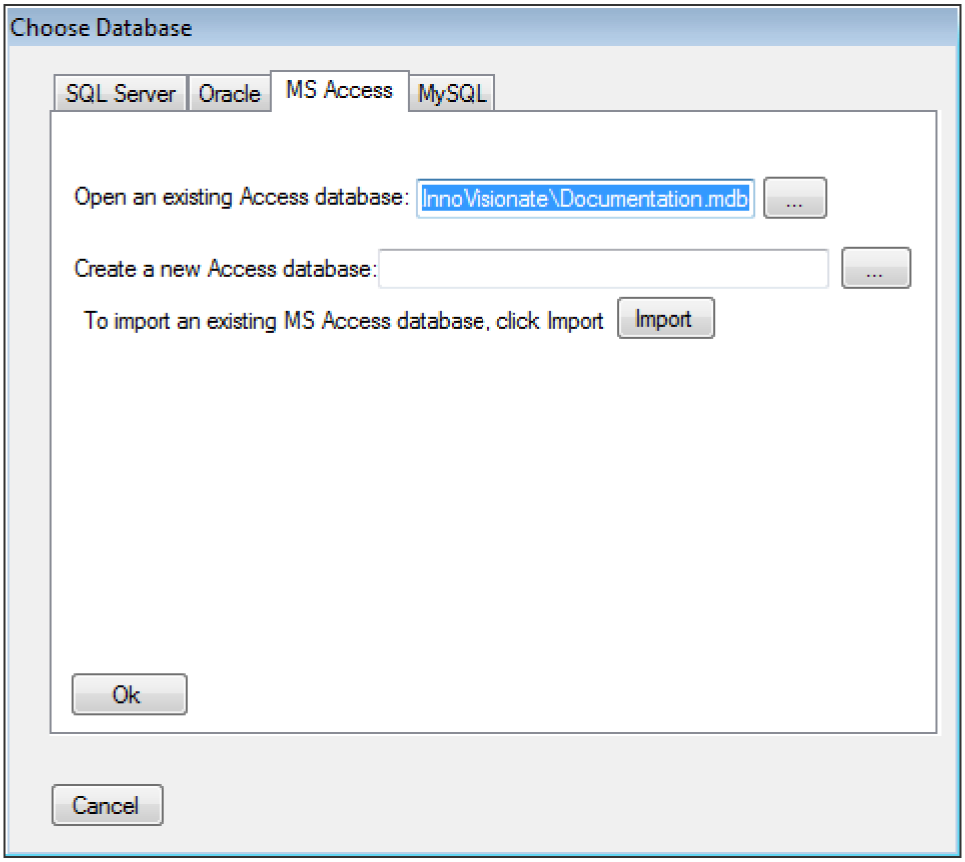
> **Note:** The use of MS Access as the database does *not* require the installation of any Microsoft Office product or licensing on the client machine, the software used to utilize Access is provided free of charge in the .NET Framework.

Visio Logger can even *migrate* data among these databases. This allows you to, for example, create your database in SQL Server, migrate it down to an Access Database file and e-mail that database to another facility to allow them to migrate it up into any of the other database systems.

Choosing a database

When Visio Logger is run by an administrator (right click the Visio Logger icon and choose Run as Administrator, or have IT logon and run the system), the menu options for working with the database are available.

To specify the database, select Tools → Database → Choose … from the Visio Logger main window. You will see a dialog similar to Figure 27.



Choose Database

Open an existing Access database: InnoVisionate\Documentation.mdb

Create a new Access database:

To import an existing MS Access database, click Import

Figure 27 Choosing a database

The initial tab in the above dialog will be the one related to the current database in use.

Simply select your database type in the above options, and specify the settings for connecting to that database (if any).

We will discuss each database server in the order of the tabs in Figure 27.

SQL Server database

If an SQL Server database server is visible within the facility, Visio Logger can use that server.

Note that SQL Server licensing is not sold with Visio Logger, and that it is up to each facility using the system to have the appropriate license to access the server. There are different ways that Microsoft allows client machines to utilize SQL Server, but it is understood that in many cases, if a facility already has a server, and other applications running on the client computers can already use that server, it is likely that the facility's license also allows use of that server by Visio Logger.

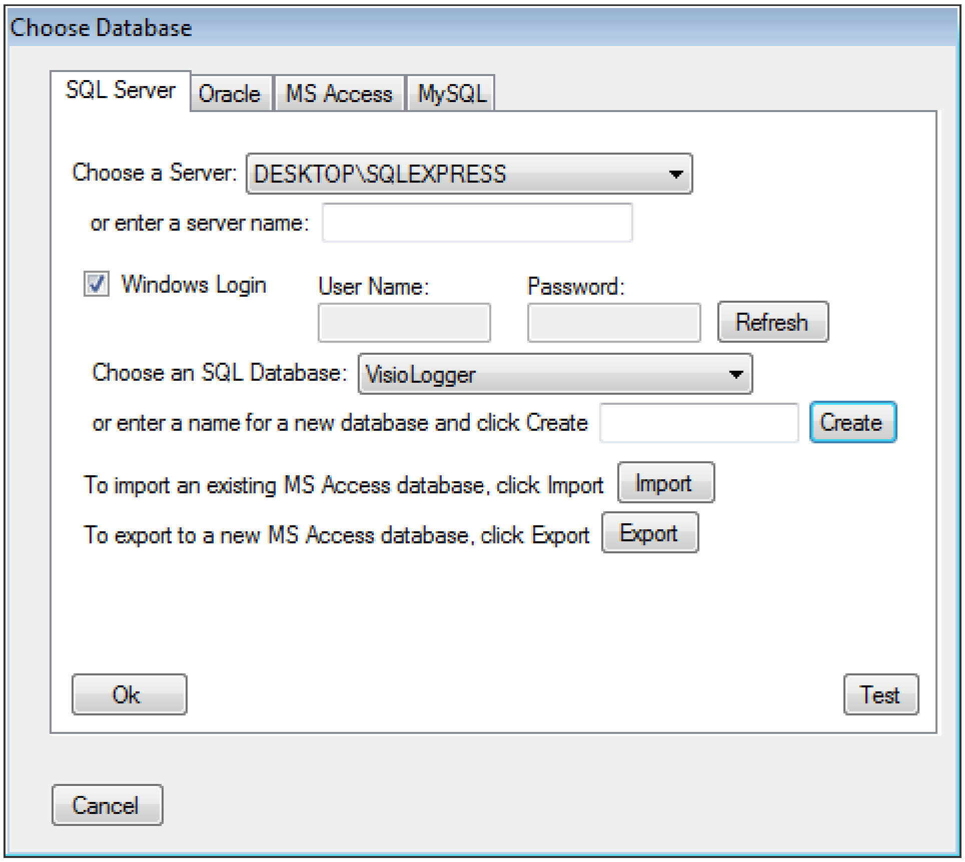
SQL Server User Credentials

There may be two different types of credentials used in accessing the Visio Logger database on SQL Server. At some point, the database has to be created and the fields for the desired data to be captured defined. To create the database, "create database" rights must be available to the person operating the dialog shown and described below. To define fields (described at length in prior sections), someone with “create columns” privileges must be using the Visio Logger system during fields definition (Edit → Columns described in prior sections). And finally, for day to day operation, someone with “insert rows” privileges must be logged in. This is accounted for by the credentials provided when the choice of the database is made (Figure 28), for normal operation, set these credentials such that they authenticate a user with at least insert rows privileges before dispatching the dialog.

When you visit the tab labeled “SQL Server” in Figure 27, Visio Logger will attempt to discover SQL Server databases available on your network.

Note that the ability to find servers depends on certain services running on the Server machines that “broadcast” available servers. The particular computer running Visio Logger may not be able to see the server you are interested in. If this is the case, you may need to manually enter the server and sometimes the SQL Server *instance* name of the server you want to access.

Note in Figure 28 that the system *was* able to discover an SQL Server instance available, therefore, it could be chosen from the server list. However, if this server was *not* found, one would have to *enter* the server name into the field underneath this list. This entry could contain either the computer *name* of the server, if it is resolvable via DNS, or, it can contain the actual IP Address. Whether or not the backslash (“\”) and server instance name is required may depend on whether multiple server services are running on that machine.


Choose Database

SQL Server | Oracle | MS Access | MySQL

Choose a Server: DESKTOP\SQLEXPRESS

or enter a server name:

☑ Windows Login User Name: Password:

Choose an SQL Database: VisioLogger

or enter a name for a new database and click Create

To import an existing MS Access database, click Import

To export to a new MS Access database, click Export

Figure 28 Using an SQL Server database

Note in the figure that a server has been chosen, in this case the DESKTOP server running the SQLEXPRESS instance. Once a server is found, the list of databases in the middle of the figure should contain databases available on that server. If not, or if your credentials were incorrect, enter the correct credentials and click “Refresh”.

The credentials used to access the database are provided *either* by authenticating to the server with the currently logged in windows user *or* by providing a specific server defined username and password. Which of these options to use would be based on current local security policy.

To create a database, a user with sufficient privileges needs to access the server. Note that these are *database* privileges, not necessarily network or specific computer privileges (administrative).

Note that both the User Name and Password values *are* remembered in the Visio Logger run time configuration information (if the Windows Login option is *not* used) because Visio Logger must authenticate to the server every time it runs. However, we also remind the reader that these credentials can be switched to a user with simple “insert rows” privileges *after* all of the setup and configuration has been completed.

We also mention that the password is stored in the system’s run time configuration file with 128-bit encryption.

Using an existing SQL Server database

If the server selection is right, and the appropriate database exists and is selected, simply clicking OK in Figure 28 causes the system to begin using that database. You can also click Test to ensure the connection will be successful.

Creating a new SQL Server database

To create a new database, enter the name of the database desired in the space provided, and click Create. If successful, you will soon get a message to that affect and then clicking OK will choose that database and you will need to create a process and the required fields definitions as described in prior sections[12].

When you create a new Visio Logger database, you need to specify at least one Signature Process for it. Please see “When a new database is created” below which pertains to these steps when a new database is created for any of the server types.

Other SQL Server options

Note that you can import to, and export from, SQL Server databases into Microsoft Access databases. This can be useful, for example, to work with an Access database to perform data analysis on your SQL Server data on a periodic basis. Specifically, you can export into Access and then connect to Access with Microsoft Excel for detailed analysis and reporting.

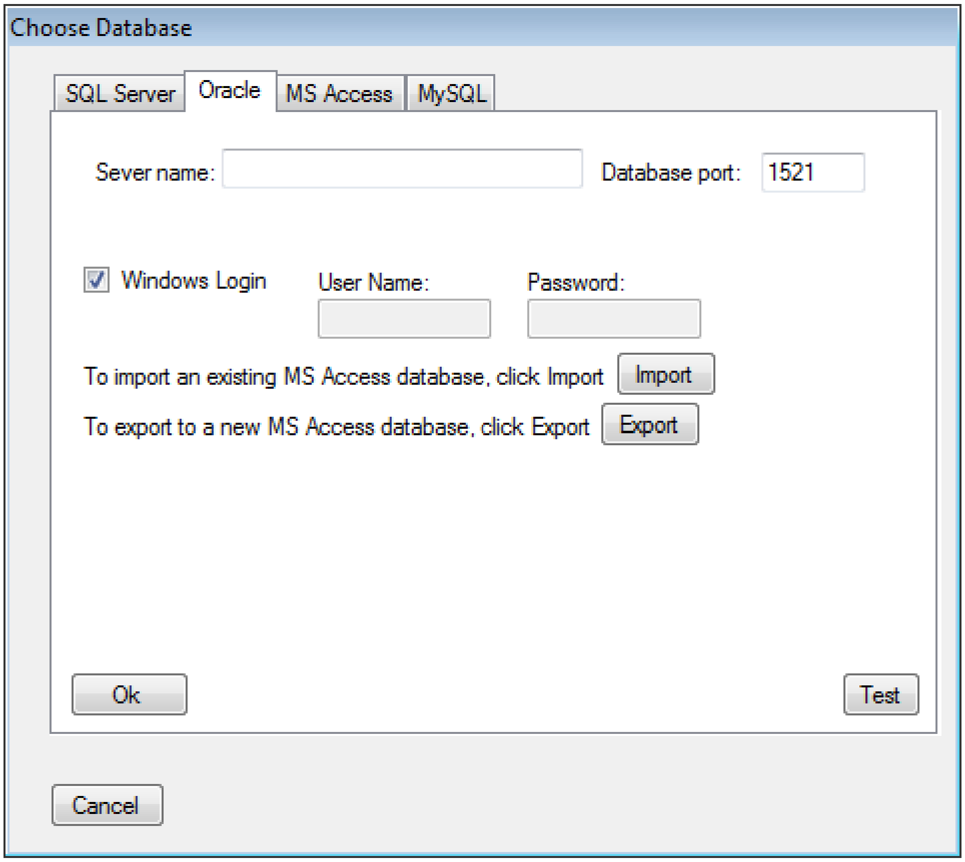
[12] Or, if you anticipate have to utilize multiple databases in your enterprise, consider defining your fields in an MS Access database, and then Import this database into your SQL Server databases.

In addition, you can easily replicate your SQL Server hosted Visio Logger system, including structure *and* data if desired, into any other SQL Server based system by simple export then import.

Oracle Databases

Oracle database selection is very similar to SQL Server selection except that there is no facility to "discover" existing databases.

Additionally, Visio Logger cannot create an Oracle database. It is necessary that the database be running on the named server (whose name you must provide) and is using the specified port, which you also must provide. Figure 29 shows the fields you would set to access an Oracle database. Remember to use the Test button to verify that your access settings are correct.



Sever name:

Database port: 1521

☑ Windows Login

User Name:

Password:

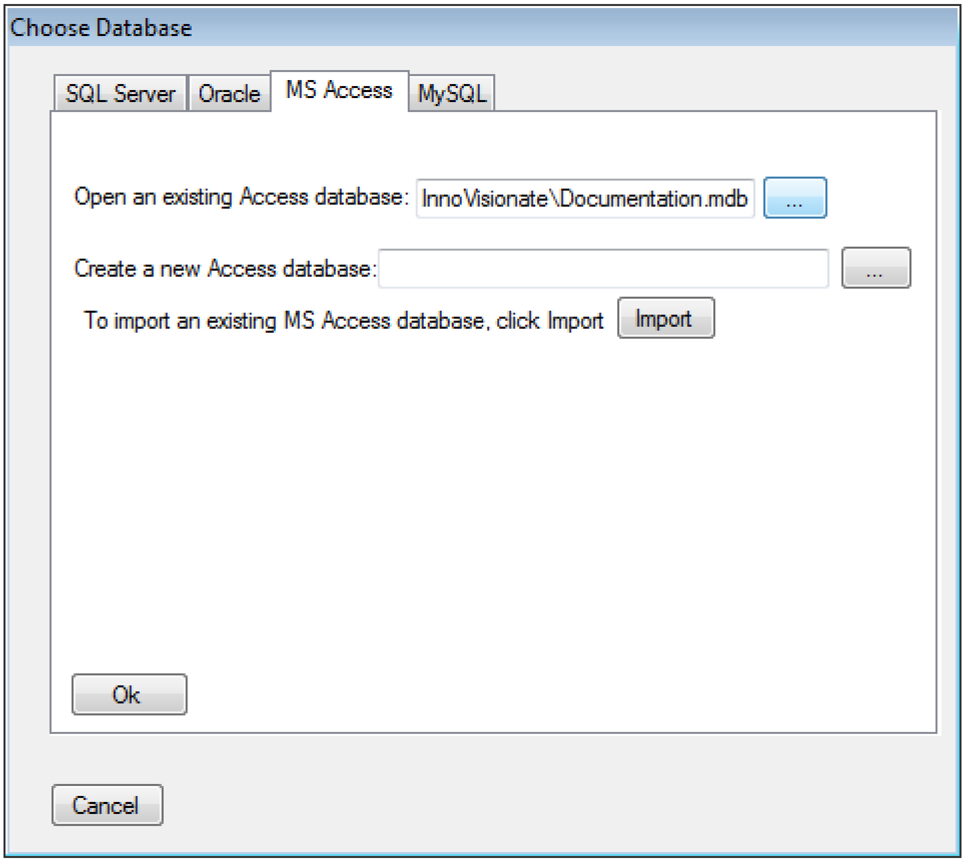
To import an existing MS Access database, click Import

To export to a new MS Access database, click Export

Figure 29 Choosing an Oracle database

Microsoft Access Databases

Using an Access database is as simple as specifying the name of the file containing the database. Note that, for multi-user installations of Visio Logger, this file must be visible by all computers running the system by placing it on a network share of some type visible to all.


Choose Database

Open an existing Access database: InnoVisionate\Documentation.mdb

Create a new Access database:

To import an existing MS Access database, click Import

Figure 30 Using a Microsoft Access database

Figure 30 shows how to specify the Access database file. Press the "…" button to the right of the existing database field to navigate to the file.

The default location for databases, specifically, the empty database installed with Visio Logger, is:

C:\ProgramData\InnoVisioNate

on Windows Vista and Windows 7, or:

C:\Documents and Settings\All Users\Application Data\InnoVisioNate

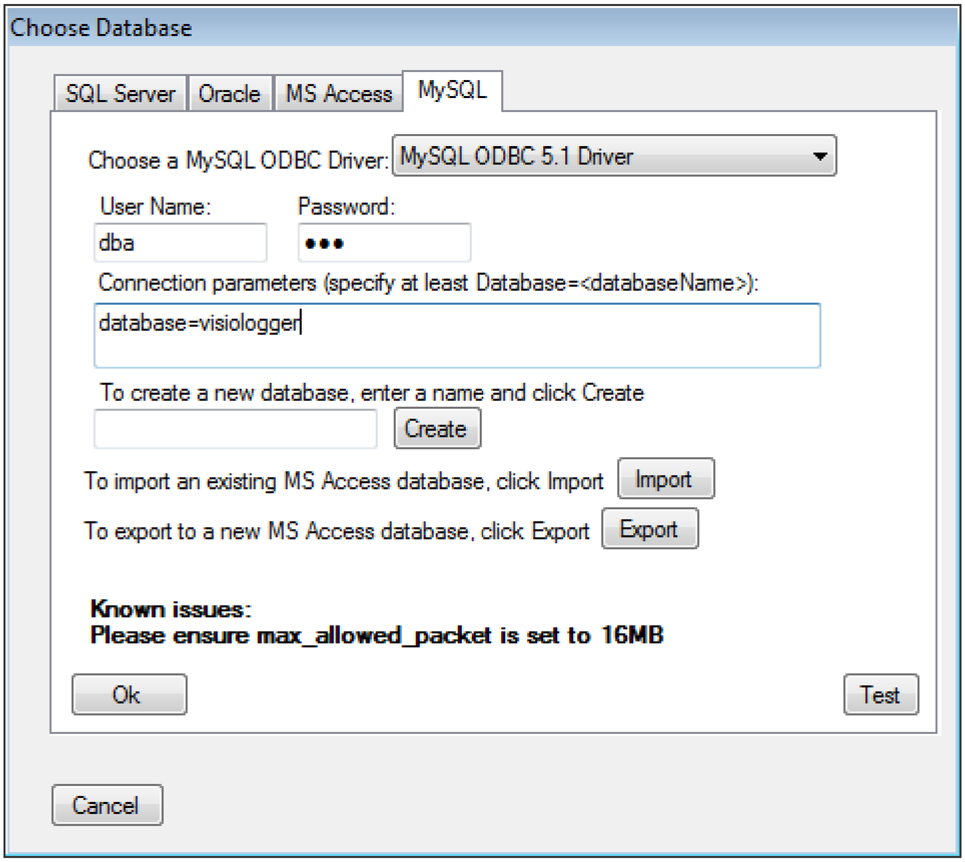
on Windows XP. However, for a sharable database, you should use the Create option and navigate to a location where all computers can read and write to the network.

Note that the "VisioLogger–Empty.mdb" file that is installed with the system should not be deleted (or used) since creating a new database is simply the act of copying this file to the desired location. If, however, you intend to create a "template" set of fields that you would then like to replicate to other systems by just copying the database. You can use the VisioLogger-Empty database and then define your pages and fields. This would create the desired template for you. However, you may also want to keep a copy of this in case, for some reason, Visio Logger needs to be re-installed and there may be a risk of it getting overwritten.

MySQL Databases

MySQL servers are accessed via ODBC from within Visio Logger.

Therefore, you need to install the MySQL ODBC Driver. Additionally, your MySQL administrator needs to provide a username and password for you to use in order to access the server.



Choose a MySQL ODBC Driver: MySQL ODBC 5.1 Driver

User Name: dba

Password: •••

Connection parameters (specify at least Database=<databaseName>): database=visiologger

To create a new database, enter a name and click Create

To import an existing MS Access database, click Import

To export to a new MS Access database, click Export

Known issues:
Please ensure max_allowed_packet is set to 16MB

Figure 31 Setting up a MySQL Server connection

Note in Figure 31 above that it is necessary to provide the "database=" connection parameter for successful connection to the database.

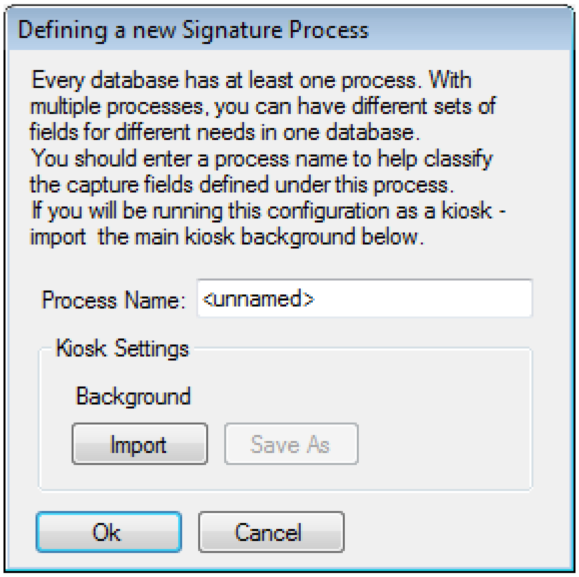
Also note the issue reported at the bottom of the dialog. This is referring to a setting in the mysql.ini file that is present on the MySQL Server computer, not to any setting in the local ODBC driver instance for MySQL.

When a new database is created

Whenever a new database is created, using any of the server technologies, Visio Logger will not have any Processes defined.

> **Note**: The exception to this is when an Access database is created, the template database used to create the new database *does* have a process named "Sign-In" defined within it. You can, however, rename this process as you see fit.

In this case, upon pressing Ok on the Choose Database Dialog, the system will then show the "New Process" dialog as in Figure 32 to allow you to create the process.



Defining a new Signature Process

Every database has at least one process. With multiple processes, you can have different sets of fields for different needs in one database.
You should enter a process name to help classify the capture fields defined under this process.
If you will be running this configuration as a kiosk - import the main kiosk background below.

Process Name: <unnamed>

Figure 32 Creating a new process in an empty database

Simply provide some name for the process and the system will apply that name to it (If you don't override the name, it will continue to be called <unnamed>). You can now define pages and fields in order to start collecting data.

Database Integrity and Protection

Visio Logger recognizes that the information gathered and maintained in your databases is critically important to you and needs to be protected from unauthorized access and information loss.

In the case of SQL Server or Oracle, it is assumed that the policies and procedures in place are there for exactly these reasons. And that the data is backed up as appropriate at the server level by IT Staff on some periodic basis. Visio Logger does not assume nor dictate what these procedures or backup intervals might be – but integrates with these by using whatever database authorization policy is in effect at your facility.

For Microsoft Access databases – it is expected that the users, or support staff, will periodically backup the network location where the database is stored (or, the local machine for a standalone database). As a less robust mechanism, Visio Logger does offer manual database "archiving" with a simple menu option. However, this technique is not intended to replace any formal policy that should be in place at your facility.

To archive a Microsoft Access database, choose Tools → Database → Archive… from the Visio Logger main menu (running with administrative privileges).

This will raise a simple dialog where you navigate to the location where archives should be maintained. Upon selecting this location, Visio Logger will then copy the MS Access database currently in use to this location, and appends the date and time to the file name. This is simply a “snapshot” of your database at this particular point in time and, again, is not intended as a formal backup procedure.

Integrating other software with Visio Logger

In the same way that Visio Logger does not assume anything about the fields and/or data that you want to capture – Visio Logger also assumes nothing about your environment in terms of existing systems, procedures, or policies.

And, just as you are utilizing your own data needs – you can also leverage your existing systems to both receive the data captured by Visio Logger *and* provide data *to* Visio Logger – in ways that can make the Visio Logger system enhance your existing systems just as if it were a native part of those systems.

Visio Logger is truly unique in this regard. We previously mentioned ”architecture”, and this is precisely what we were referring to. The built in ways that you can extend Visio Logger to become the system that can be every bit as powerful as the huge, and hugely expensive, vertical systems, that are typically out of reach for most small, or at least, economically smart, enterprises.

With Visio Logger integration, what you truly want is now within reach!

This section details how this works and how to provide the “glue” that integrates Visio Logger into your system.

Integration Architecture

Visio Logger is a Microsoft .NET Framework system built using the C# language.

At its core – the Visio Logger system is specifically designed to be able to use “other” software with just a few configuration steps required.

The machinery used to achieve this integration is the Component Object Model (COM) and consists of a series of formally defined interfaces that Visio Logger uses at specific places within its processing to connect to and utilize these external agents.

The definition of these interfaces follows; however, for interested parties who want to see how this works by inspecting real software that is already integrated into Visio Logger, note that the source code for some integrated objects is provided with Visio Logger (projects can be built using MS Visual Studio 2008 or higher). To find this source, locate the Visio Logger run time data directory, typically in:

C:\ProgramData\InnoVisionate

and navigate to the” Integration Source” sub-directory.

Interfaces

Integration happens at 6 distinct locations or times in the processing of a sign in session, which is the capture of information from any one person using the system.

> **Note:** The "events" described here are occurring on the computer that is collecting data, the "Logger" computer. In other words, even though a Viewer machine will *see* the new record appear when someone has signed in, the "New Record" integration steps do not occur on *that* computer, they have already occurred on the Logger computer.

The only exception to the above note is the Action event that is described below. This is by design because the Action interface is specifically provided to allow a user at any computer to press a button in the sign-in data to cause something to happen. However, for the other interfaces, it would not be appropriate to have every machine on the network performing these tasks – that should be left to the one (or any) computer at which the data originates.

The points in the process where these interfaces are used are:

1. Pre-Signature: When the system "cycles" – that is, at the point where it is just beginning the sign in process for an individual.

   You can use this opportunity to pre-load the information in the Visio Logger database for the next signer's benefit. For example, if you have a Multi-Valued options field – you could set the available options that are pertinent for that particular day or time of day.

   Note that this interface causes the row to be created for the signer (populating the Sign In time cell automatically). One *can* pre populate the values of the fields, however, at this point, no entry has been made by the signer, so there is no information available to obtain external information about that particular signer.

   However, there is another useful reason to utilize this interface. With it, you can launch a Windows Forms object at this point and use that to capture *all* of the input for your signer. If you also make all of your fields "ignorable" you have effectively replaced the entire Visio Logger sign-in user interface mechanism with your own user interface, which you can customize however you see fit.

2. When a Page is submitted. Recall that a page is a collection of entry fields submitted to the visitor as one "entry" session in the flow of the sign in process.

   Typically a page is one field, when fields are either signature pad keyboard entry, or menu fields.

   In any case, every time such a page is submitted by the action of the visitor, all of the integration tools specified to be used for this event are called.

   This particular event is useful in directing the flow of your sign in process across multiple paths of input requirements. For example, when a particular field is provided through this event with a particular value, you can then set the value of *other* fields, which means that *those* fields can be skipped through the remainder of the session for this signer.

A powerful and configurable “flow” manager is included with Visio Logger out of the box that relies heavily on this event. With that tool, you can configure Visio Logger to collect entirely different sets of the data based on the value of any field that they’ve previously provided. To learn more about it, please reference the “Using the InnoVisioNate Page Submitted Processor” document in the InnoVisioNate ProgramData directory, or from the InnoVisioNate website.

3. Record Changed: Some fields can be changed by a user, for example “Editable” fields. When this happens, you may want some external system to record the fact, or you may want to set the value of one of your fields based on the new value of a field.

   Additionally, when Actions occur, the row of data *also* changes, and this event is fired. Do not forget that this is in *addition* to the Action event described next.

   You would use this integration event to accomplish these tasks.

4. When an Action occurs: Your sign-in data may include any number of Action fields that are buttons in the rows of data until someone presses that button.

   Use an “action” event integration object to notify an external system of some event or condition about this particular sign-in data, or to modify existing data in the current record. Overall, Actions occur only when staff specifically decides that it should by pressing the button.

   A very good example of this event is the implementation of the InnoVisioNate Video tools. Here, you can define an action field in your data that might, for example, take a picture of a person or object somewhere or sometime *after* they have completed the sign in.

5. After the record is complete: When the signer completes the last page of data entry, all the data has been entered for that record and this interface will be used. At this point, you could contact an external system and pass on all the information about the signer. You could also populate other fields in the sign-in data. Probably fields that are “ignorable” which don’t require entry by the signer but which may be useful to accompany the other data in the record.

   An excellent example of this information is provided with Visio Logger in the form of the “InnoVisioNate Sign Out Processor”. This tool takes the data submitted from the “new” sign in record, and actually *finds* a related sign IN record by using information in the new record. It then *deletes* the new record, and populates certain fields in the *found* Sign In record. Thus providing powerful and configurable Sign Out capability with Visio Logger. This is in addition to the system’s simpler Auto Logger mode that provides Sign In AND Out capabilities for less complex data environments. With this event, however, the system can *still* use the more robust Logger mode of operation, yet allow the facility to achieve true Sign in AND out within that mode.

6. When a key is tapped on the signature pad. This event gives your software control as each letter is added to the current "result" in a pad keyboard entry field. The possibilities here are limitless, but include the capability to, as an example, "AutoFill" your data by inspecting the current value against any external or internal data.

Data flow

All of the interfaces described follow the same basic structure in terms of receiving data from Visio Logger, or providing data back to the system.

In particular, field names, field values, image field names, and handles to images are all passed back and forth between Visio Logger and any external object that it utilizes.

All of these are arrays of data, in a C# object, they are System.Array values, in other contexts, they can be thought of as SAFEARRAYS. The names and field values are string arrays, and any "images" are passed as handles to the image(s), which the external object can use to either recreate the image for its own purpose, or to provide an image back to Visio Logger to have it placed in its database.

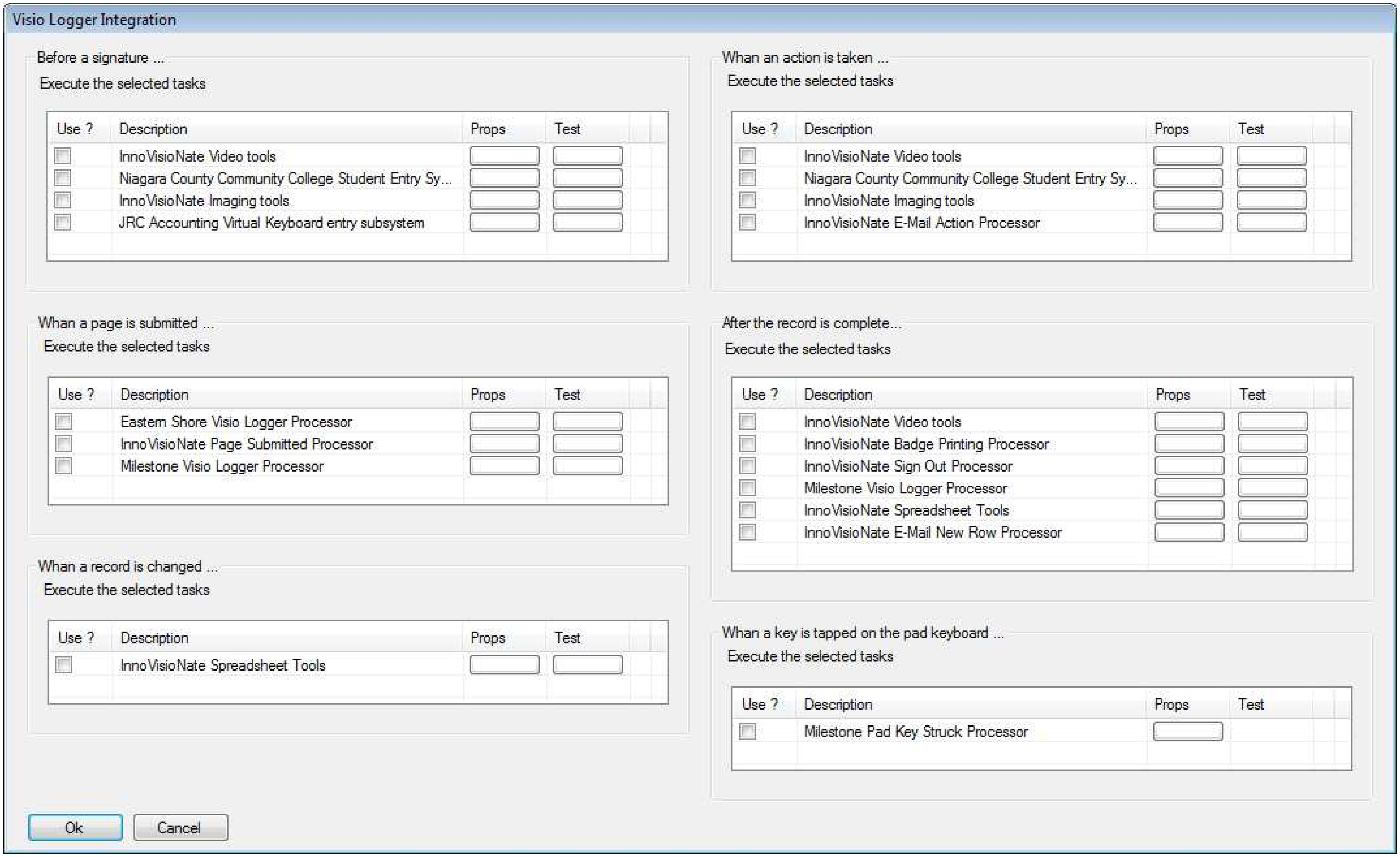
With minor exceptions, all of these interfaces operate this way.

Additional information provided includes database information for the hosting Visio Logger database in the event that the external object needs information about the current Visio Logger process in order to do its job.

Configuring Integration

At the process level, you can choose which of the available integration objects will be used and when. Recall that process level means at the level that contains all the fields desired for a particular type of sign-in system. You can have multiple processes, for multiple facilities, for example, and you can define different choices for the configured integration objects for each process.

If, again running with administrative privileges, you choose Tools → Toolbox… from the Visio Logger main menu, you will see a dialog similar to the following. Note that your particular view will only contain integration tools shipped with Visio Logger, the provided image is from a computer with other custom tools installed.



Visio Logger Integration

Before a signature ...

Execute the selected tasks

Use ?	Description	Props	Test
☐	InnoVisioNate Video tools		
☐	Niagara County Community College Student Entry Sy...		
☐	InnoVisioNate Imaging tools		
☐	JRC Accounting Virtual Keyboard entry subsystem		

Whan a page is submitted ...

Execute the selected tasks

Use ?	Description	Props	Test
☐	Eastern Shore Visio Logger Processor		
☐	InnoVisioNate Page Submitted Processor		
☐	Milestone Visio Logger Processor		

Whan a record is changed ...

Execute the selected tasks

Use ?	Description	Props	Test
☐	InnoVisioNate Spreadsheet Tools		

Whan an action is taken ...

Execute the selected tasks

Use ?	Description	Props	Test
☐	InnoVisioNate Video tools		
☐	Niagara County Community College Student Entry Sy...		
☐	InnoVisioNate Imaging tools		
☐	InnoVisioNate E-Mail Action Processor		

After the record is complete...

Execute the selected tasks

Use ?	Description	Props	Test
☐	InnoVisioNate Video tools		
☐	InnoVisioNate Badge Printing Processor		
☐	InnoVisioNate Sign Out Processor		
☐	Milestone Visio Logger Processor		
☐	InnoVisioNate Spreadsheet Tools		
☐	InnoVisioNate E-Mail New Row Processor		

Whan a key is tapped on the pad keyboard ...

Execute the selected tasks

Use ?	Description	Props	Test
☐	Milestone Pad Key Struck Processor		

Figure 33 Installed Integration tools

The six lists shown in Figure 33 are organized by the 6 interface types previously described.

> **Note:** The settings indicating (the “Use ?” checkboxes) which integration objects to run is part of the *process*, and is therefore stored in the particular database being used. Though the selection of what tool(s) to run does not have to be made on every computer using Visio Logger – be aware that each tool *may* at its option, store its own configuration settings on the local computer. This may impact whether the tool needs to be configured on a per computer basis.
>
> There are 2 types of configuration – specifying the “Use ?” value as in Figure 33, which is stored in the database, can be done on any computer.
>
> However, the “Props” configuration is for *that* tool – and those should generally be configured on the Logger computer(s), except for the Action interface object – which should be configured on any computer that might press the button that would use the action.

Note that an item appears in Figure 33 by virtue of it having been “registered” as supporting one or more of the 6 interfaces published by Visio Logger. The source code provided with Visio Logger for some of the objects shown in the figure contain information that shows how to do this.

Simply indicate which of these objects will be used by checking the “Use ?” checkbox on the left side of each option. If you want to configure any object simply press the button in the “Props” column. Each object implements its own properties and is responsible for providing a user interface and storage mechanism to allow the setting of these properties.

To see how the object works, or during build and test of the object, click the button in the Test column for the object. In this case, the object will be working with the last row of data in the Visio Logger Data view – if there is no data, either create a row, or consider using Edit Preferences to change the date range to include a date where there is data.

Sample Integration Objects

As previously described, we have provided some source code for integration objects to help you get started with developing any new capability to work within your existing systems.

At least one of these objects is written in C++ and uses typical COM technology to present itself for use by Visio Logger. And at least one is written in C# and shows the slightly different way in which it registers itself as providing one or more of the required interfaces.

In either case – Visio Logger itself uses pure COM to talk with these objects. In other words, there is nothing about .NET that is required of your implementation, you can write them in any language you choose, .NET or not, but they do need to be COM visible and they must publish the fact that they support at least one of the Visio Logger interfaces.

Integration Tools provided with Visio Logger

Integration is such a powerful concept to increase the utility of Visio Logger – InnoVisioNate provides many integrated objects free of charge and installs them with the software. Indeed, for some, the source code is also provided in case users would like to enhance their operation.

We previously outlined some of the tools in the document introduction. Here we expand on the details of using and configuring them.

InnoVisioNate Video Tool

This object is used to snap a picture of the signer at sign in time. If the machine hosting the sign in process has access to a webcam, or other video device, this tool can activate that camera and snap a picture of whatever it is pointing to at the time. This can be done without the signer even knowing it is happening.

InnoVisioNate Scanning Tool

This object is very similar to the Video Tools – except that it uses a scanner instead of a camera to capture the image. Allowing you to include any document, insurance card, license, etc., that you may need with your data.

The source code for this tool is also provided. Because it uses the generic windows interface to scanner devices, a facility could improve the user experience if there was a specific scanner interface that could be used instead. Chief among these goals might be to pre-configure the scanner to automate the initiation of the scan without having to go through the settings every time.

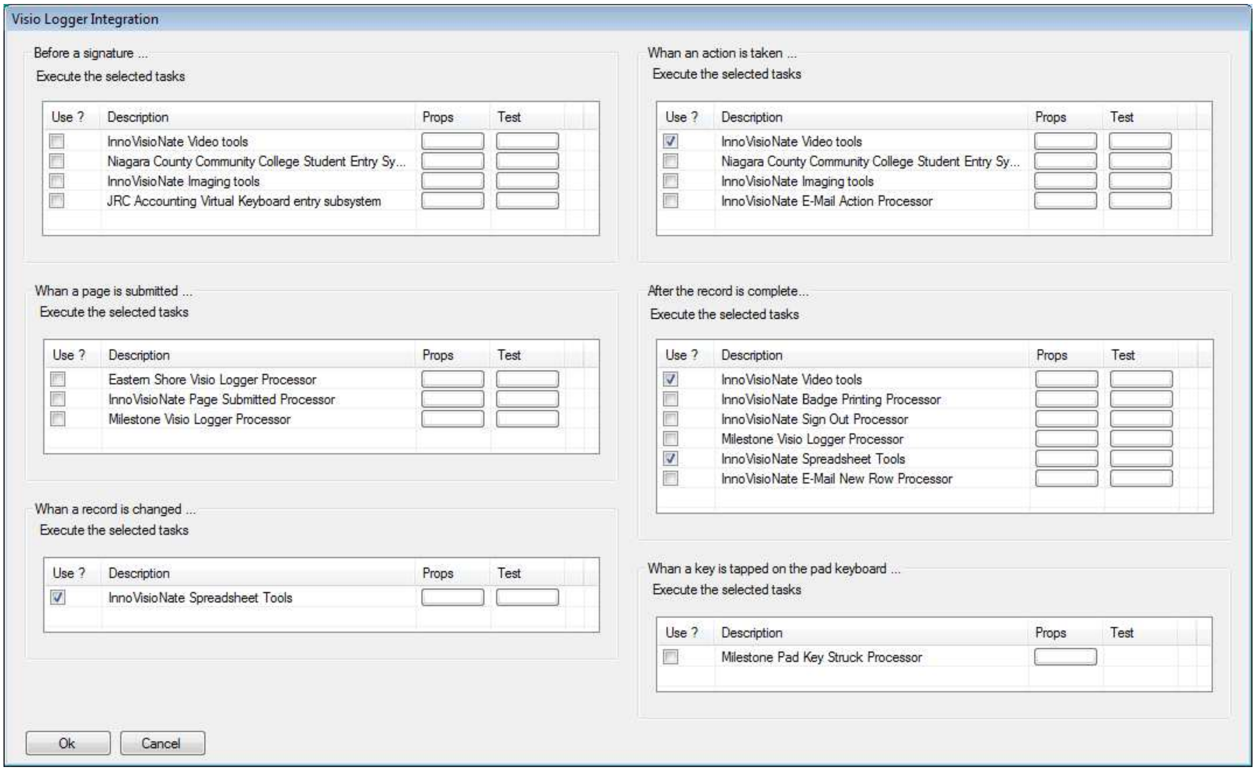
> **Note:** The full documentation for the Video and Scanning tools is provided in:
>
> > Using the InnoVisioNate Video and Scanning Tools.pdf
> >
> > Available on the Visio Logger website at: http://www.visiologger.com

InnoVisioNate Spreadsheet Tools

Spreadsheets are a powerful tool for data storage and analysis. So, why not place your sign-in data directly in one?

The InnoVisioNate Spreadsheet tools do just that – you can create an Excel Workbook, set a few simple properties regarding what you want to store in the sheets of that book – and Visio Logger will automatically populate that sheet as each signer enters data into your system. In addition, this tool can track *changes* to your sign in data.

To use this tool, select Tools → Toolbox… from the Visio Logger main menu.



Visio Logger Integration

Before a signature ...
Execute the selected tasks

Use ?	Description	Props	Test
☐	InnoVisioNate Video tools		
☐	Niagara County Community College Student Entry Sy...		
☐	InnoVisioNate Imaging tools		
☐	JRC Accounting Virtual Keyboard entry subsystem		

Whan an action is taken ...
Execute the selected tasks

Use ?	Description	Props	Test
☑	InnoVisioNate Video tools		
☐	Niagara County Community College Student Entry Sy...		
☐	InnoVisioNate Imaging tools		
☐	InnoVisioNate E-Mail Action Processor		

Whan a page is submitted ...
Execute the selected tasks

Use ?	Description	Props	Test
☐	Eastern Shore Visio Logger Processor		
☐	InnoVisioNate Page Submitted Processor		
☐	Milestone Visio Logger Processor		

After the record is complete...
Execute the selected tasks

Use ?	Description	Props	Test
☑	InnoVisioNate Video tools		
☐	InnoVisioNate Badge Printing Processor		
☐	InnoVisioNate Sign Out Processor		
☐	Milestone Visio Logger Processor		
☑	InnoVisioNate Spreadsheet Tools		
☐	InnoVisioNate E-Mail New Row Processor		

Whan a record is changed ...
Execute the selected tasks

Use ?	Description	Props	Test
☑	InnoVisioNate Spreadsheet Tools		

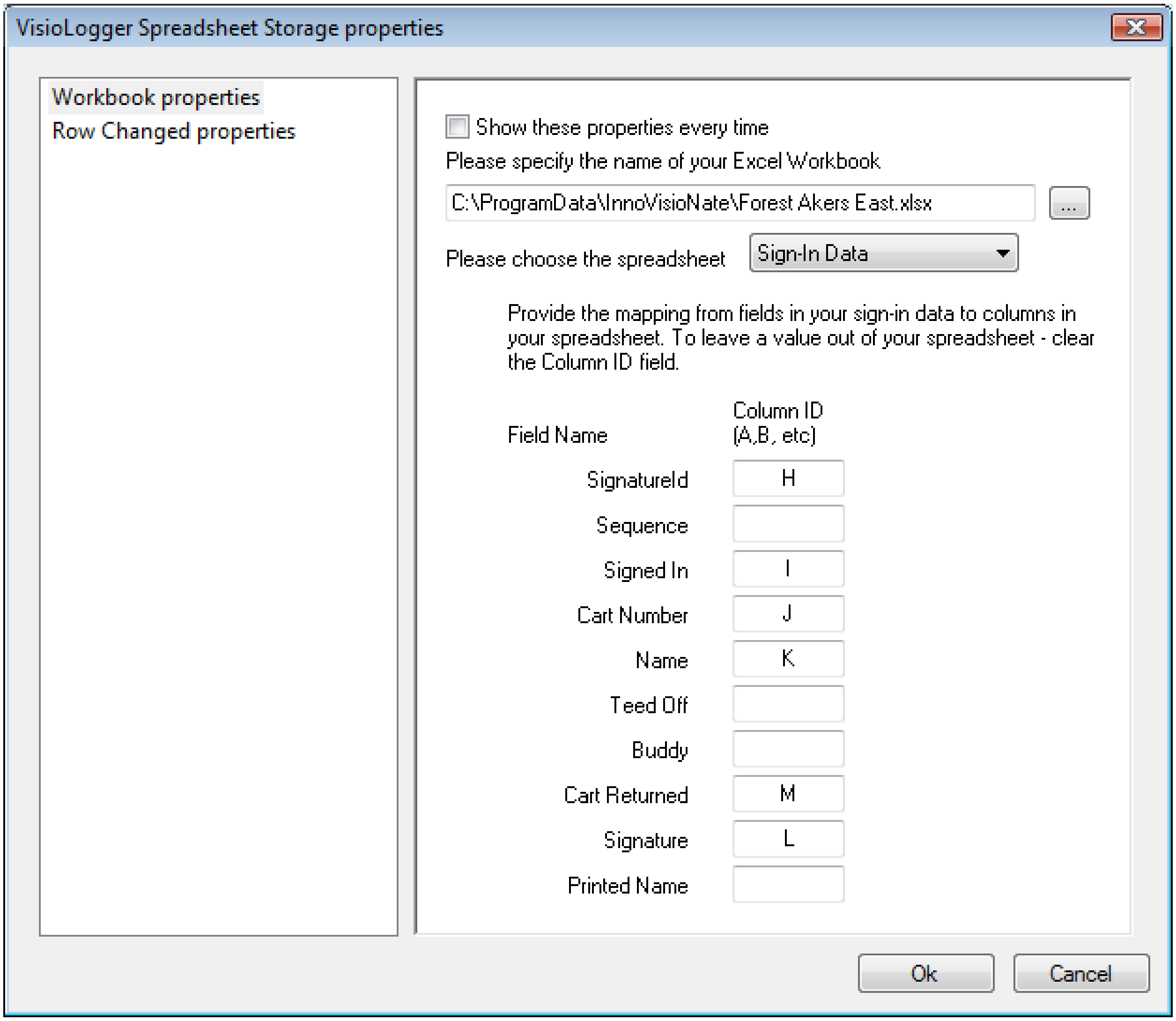
Whan a key is tapped on the pad keyboard ...
Execute the selected tasks

Use ?	Description	Props	Test
☐	Milestone Pad Key Struck Processor		

Figure 34 Turning on the Spreadsheet tools

In Figure 34, we have selected the Spreadsheet tools to fire on “After the record is complete” – and on record Changed. Other tools, such as the Video tools – can still be run without conflict.

Now, we select the Props button on the Spreadsheet Tools row:



VisioLogger Spreadsheet Storage properties

□ Show these properties every time

Please specify the name of your Excel Workbook

C:\ProgramData\InnoVisioNate\Forest Akers East.xlsx

Please choose the spreadsheet: Sign-In Data

Provide the mapping from fields in your sign-in data to columns in your spreadsheet. To leave a value out of your spreadsheet - clear the Column ID field.

Field Name	Column ID (A,B, etc)
SignatureId	H
Sequence	
Signed In	I
Cart Number	J
Name	K
Teed Off	
Buddy	
Cart Returned	M
Signature	L
Printed Name	

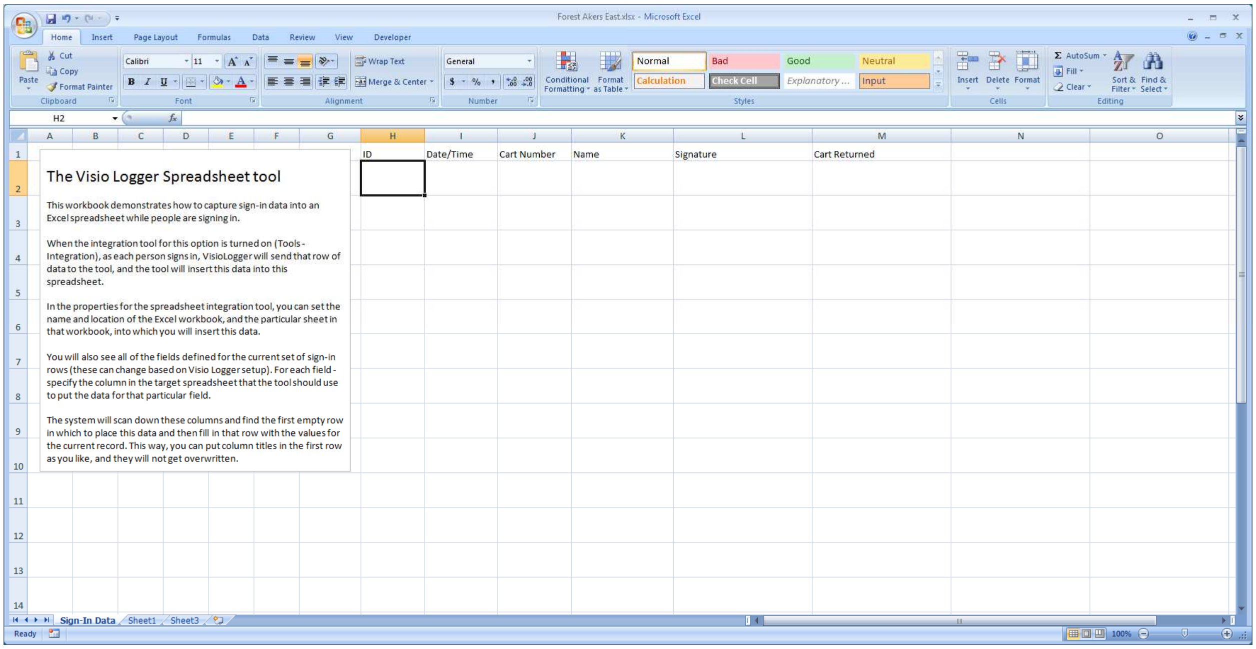
Figure 35 Spreadsheet Tools configuration

Figure 35 shows how to map the incoming fields to the spreadsheet columns. We have selected a workbook, then, from that workbook selected the particular sheet in which we will store data. And finally, for the fields we want to store, we indicate which column in that sheet will get populated with the data coming from the signer. Note carefully that we will retain the SignatureId field – this field is always provided and is the actual database ID value for this particular row of data (Sequence and Signed In are also always present).

Fields we don't want to store have no column associated with them.

We'll visit the Row Changed properties later (this particular object has its source code provided with Visio Logger and provides a good example of how an object can support multiple Visio Logger integration interfaces).

Now we will run Visio Logger and sign in a golfer, first, however, we'll show our target spreadsheet (Note that this workbook does *not* have to be open while signing is occurring – in fact, any individual at the computer may not even know this is happening):



The Visio Logger Spreadsheet tool

This workbook demonstrates how to capture sign-in data into an Excel spreadsheet while people are signing in.

When the integration tool for this option is turned on (Tools-Integration), as each person signs in, VisioLogger will send that row of data to the tool, and the tool will insert this data into this spreadsheet.

In the properties for the spreadsheet integration tool, you can set the name and location of the Excel workbook, and the particular sheet in that workbook, into which you will insert this data.

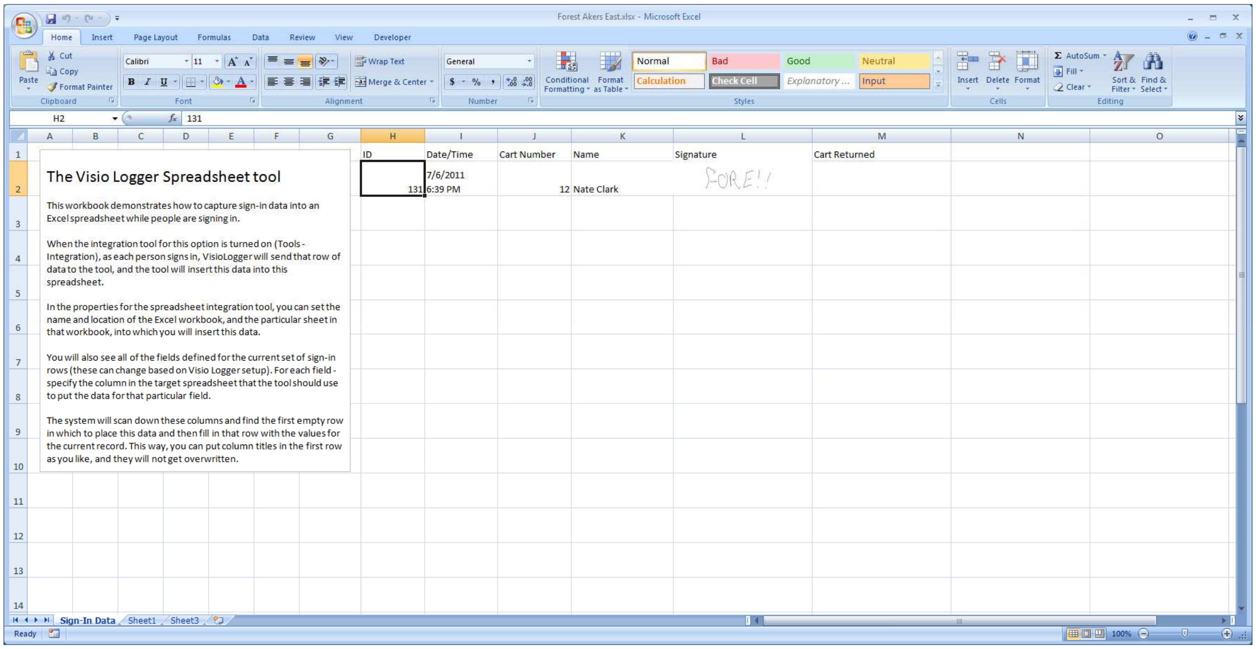
You will also see all of the fields defined for the current set of sign-in rows (these can change based on Visio Logger setup). For each field - specify the column in the target spreadsheet that the tool should use to put the data for that particular field.

The system will scan down these columns and find the first empty row in which to place this data and then fill in that row with the values for the current record. This way, you can put column titles in the first row as you like, and they will not get overwritten.

ID	Date/Time	Cart Number	Name	Signature	Cart Returned

Figure 36 Spreadsheet Tools sample Spreadsheet

Golfer Nate Clark has checked out cart # 12, the date and time, name, cart # and signature are recorded as in Figure 37. His signature indicates that he is probably a bit dangerous.



The Visio Logger Spreadsheet tool

This workbook demonstrates how to capture sign-in data into an Excel spreadsheet while people are signing in.

When the integration tool for this option is turned on (Tools-Integration), as each person signs in, VisioLogger will send that row of data to the tool, and the tool will insert this data into this spreadsheet.

In the properties for the spreadsheet integration tool, you can set the name and location of the Excel workbook, and the particular sheet in that workbook, into which you will insert this data.

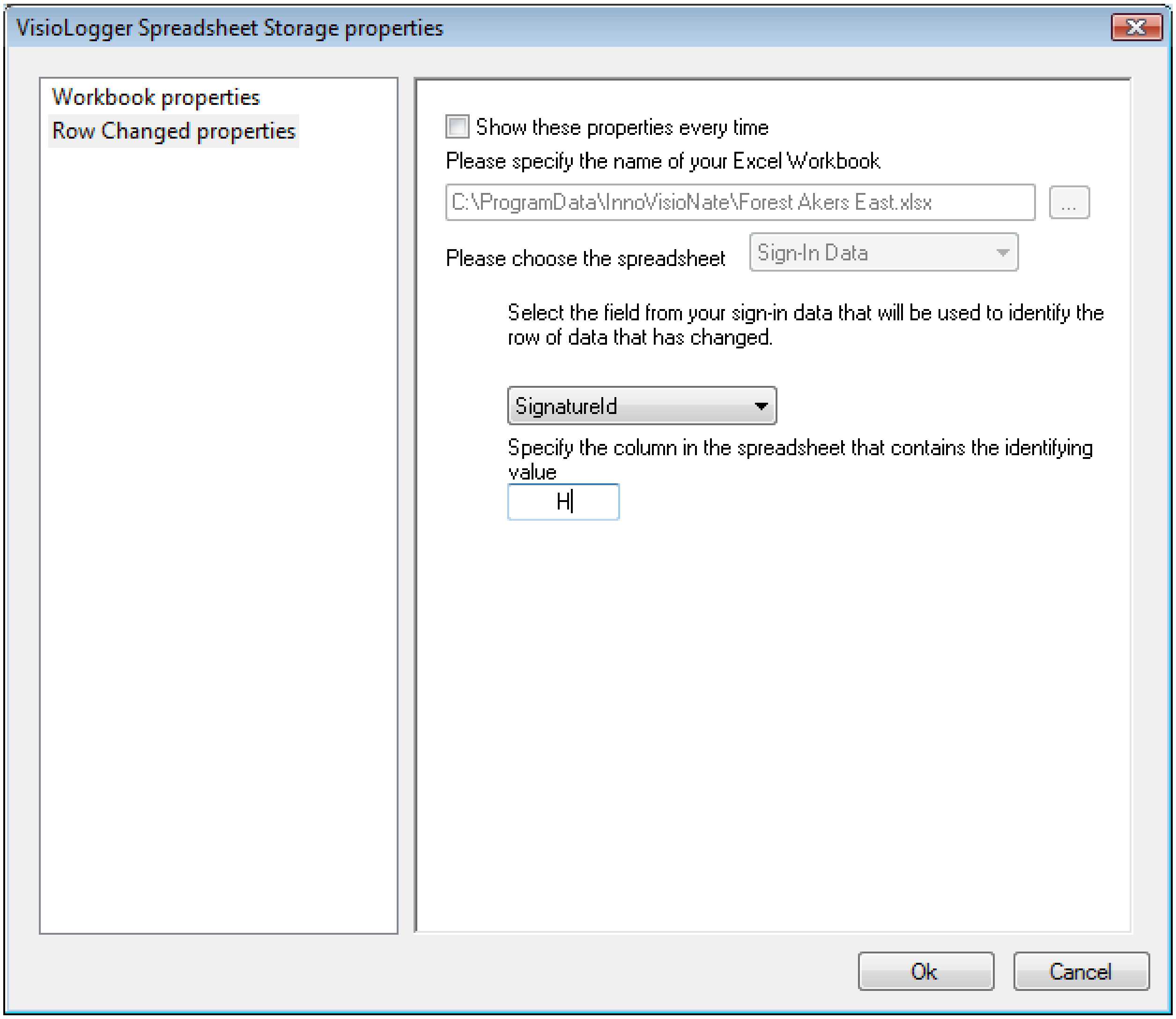
You will also see all of the fields defined for the current set of sign-in rows (these can change based on Visio Logger setup). For each field - specify the column in the target spreadsheet that the tool should use to put the data for that particular field.

The system will scan down these columns and find the first empty row in which to place this data and then fill in that row with the values for the current record. This way, you can put column titles in the first row as you like, and they will not get overwritten.

ID	Date/Time	Cart Number	Name	Signature	Cart Returned
131	7/6/2011 6:39 PM	12	Nate Clark	FORE!!	

Figure 37 A new cart sign out row

Now we can show how to leverage the Row Changed feature of this object. In our field definitions for this database – there is a Cart Returned Action. We visit the properties for the Spreadsheet tools once again, but this time change the properties for the “Row Changed” action.



VisioLogger Spreadsheet Storage properties

□ Show these properties every time

Please specify the name of your Excel Workbook

C:\ProgramData\InnoVisioNate\Forest Akers East.xlsx

Please choose the spreadsheet: Sign-In Data

Select the field from your sign-in data that will be used to identify the row of data that has changed.

SignatureId

Specify the column in the spreadsheet that contains the identifying value

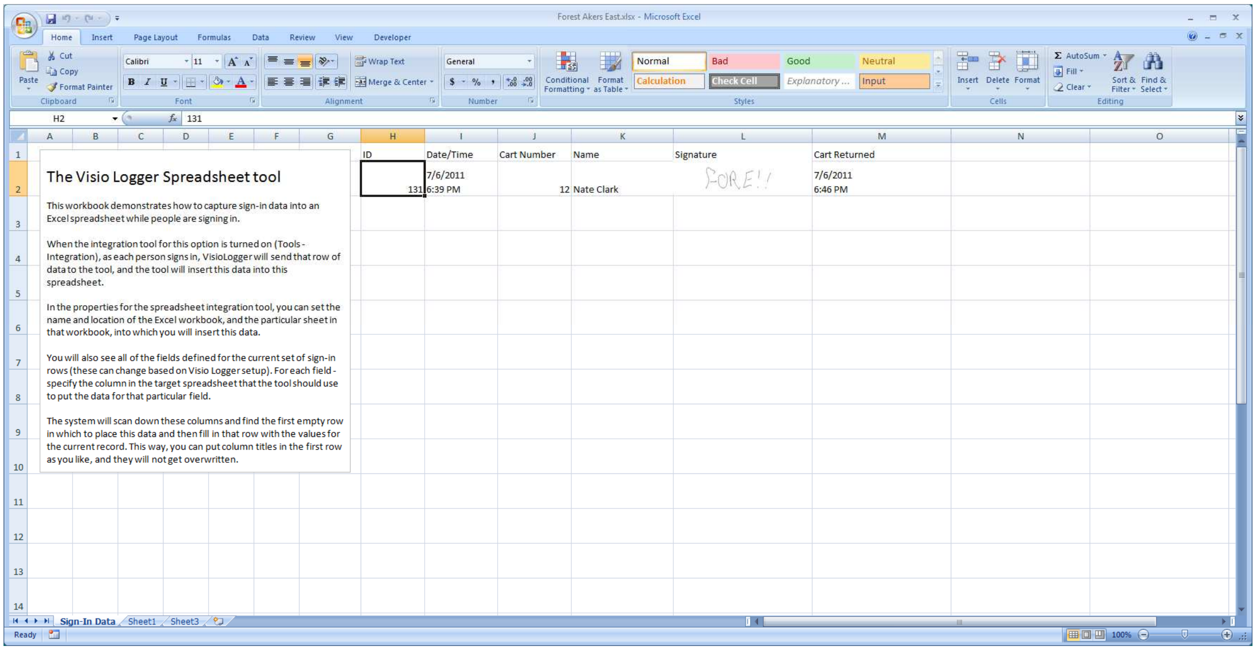
H

Figure 38 Spreadsheet Tools Change Row properties

Now we tell the Spreadsheet tool that our spreadsheet contains an identifying value in column H, and, in this case, we will use the SignatureId value of the incoming data to find the correct row. This identifying data could be anything as long as it will be unique. In particular, data entered by a golfer may not necessarily be unique, so the SignatureId (being guaranteed unique) is recommended for this purpose as it is always available to any integration object.

Since our data set includes an action column called Cart Returned, and we have mapped this value to one of our sheet’s columns, the sign in row will change when the starter indicates the cart has been returned by pressing this button.

Pressing the button (which could be done on a different computer altogether), causes our spreadsheet to look like this:



H2: 131

The Visio Logger Spreadsheet tool

This workbook demonstrates how to capture sign-in data into an Excel spreadsheet while people are signing in.

When the integration tool for this option is turned on (Tools - Integration), as each person signs in, VisioLogger will send that row of data to the tool, and the tool will insert this data into this spreadsheet.

In the properties for the spreadsheet integration tool, you can set the name and location of the Excel workbook, and the particular sheet in that workbook, into which you will insert this data.

You will also see all of the fields defined for the current set of sign-in rows (these can change based on Visio Logger setup). For each field - specify the column in the target spreadsheet that the tool should use to put the data for that particular field.

The system will scan down these columns and find the first empty row in which to place this data and then fill in that row with the values for the current record. This way, you can put column titles in the first row as you like, and they will not get overwritten.

ID	Date/Time	Cart Number	Name	Signature	Cart Returned
131	7/6/2011 6:39 PM	12	Nate Clark	FORE!!	7/6/2011 6:46 PM

Figure 39 Update spreadsheet data

Therefore, this golfer checked out cart # 12 at 6:39 pm, and returned it at 6:46 pm, a pretty quick round of golf or a very fast cart.

Issues that can occur with Excel

It is important to note a few issues that can happen when working with Spreadsheet integration.

First, the workbook should not be open over the network by any other person when Visio Logger attempts to access it. It is ok for the workbook to be open on the computer running Visio Logger, though the user at that computer would need to be sure and save the workbook when they were done with it.

Also – Excel integration will not work if a user (on the Visio Logger computer) has any workbook open and is currently editing a formula in the formula bar. Excel will silently ignore the attempts by Visio Logger to open and/or work with its spreadsheet whenever it is busy. To see an indication of this, note that when you click in the Excel formula bar, the Excel menu system becomes disabled, whenever this is the case, Excel integration with other software will not succeed.