



Using the InnoVisioNate Page Submitted Processor

---

**Table of Contents**

- Visio Logger can take different “paths” through your data fields..... 3
- Launching the Page Submitted Processor Properties..... 3
- Validation Settings..... 4
- Creating Value Setting rules..... 6
  - When Signature (bitmap) fields are the target ..... 11
- Saving the Configuration..... 11
- Summary..... 11

# What is the Page Submitted Processor?

---

## Visio Logger can take different “paths” through your data fields

When you collect different data from visitors based on why they are there, or for any other reason, Visio Logger can present only those “pages” of data entry fields relevant for *that* visitor.

Recall that a page is the presentation of a certain set of fields grouped together. There can be multiple entry fields on a single page. However, when a signature pad is being used, and a particular field uses either the pad keyboard, or a menu choice, there can be only one data field per page.

The “series” of pages happens based on the order that you define them in. After defining them, you can shift pages to the left and re-order them in any way desired.

Essentially, Visio Logger starts with the first page for each new signer, and goes through them in order after the visitor enters data for each field.

If, when Visio Logger encounters a page whose data entry fields are *already* filled in, it silently *skips* that page.

Therefore, the way that the Page Submitted Processor works is to inspect the data after each page is submitted, and, based on the values of the fields on that page, actually *sets* the values of any *other* fields to a configurable value.

This not only allows subsequent pages of data to be skipped, you are also provided the opportunity to *automatically* populate fields based on the value of any given field.

## Launching the Page Submitted Processor Properties

When Visio Logger is running under a user with administrative privileges on the computer, the main menu will contain a Tools → Toolbox entry. Select this option and find the InnoVisioNate Page Submitted Processor in the “When a page is submitted ...” group in the middle left of the Toolbox.

The Props column contains a button, which when pressed, presents a dialog similar to Figure 1.

Note that the figure shows Current process and database information at the top. This is indicating that these settings are for the particular process and database that is in use when you show these properties. It is important to remember that the settings defined are for *that* database. Therefore, if you have multiple databases, or even multiple processes within a database, the properties you set will not confuse those from another database.

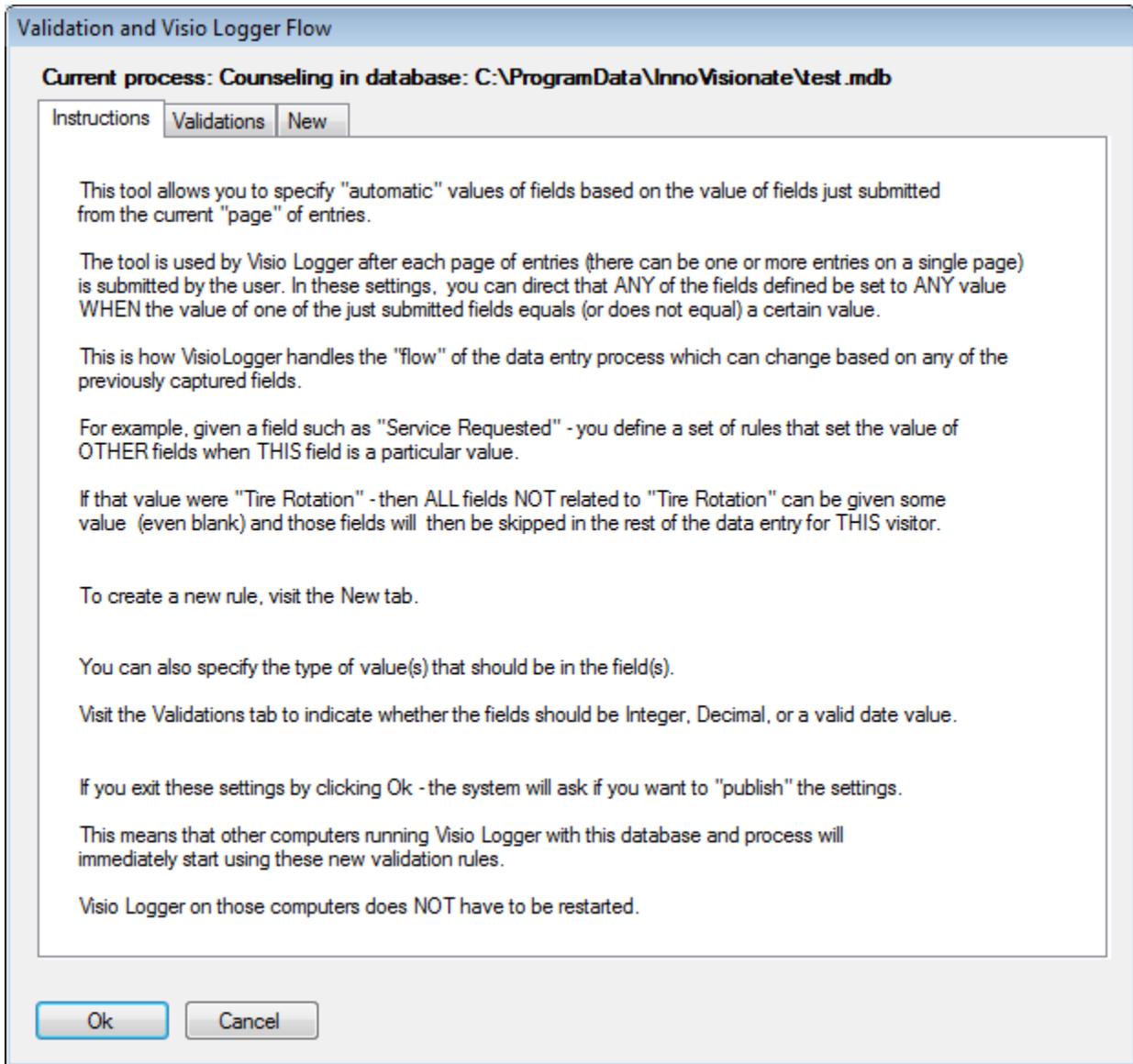


Figure 1 Page Submitted Properties Dialog

There are two types of properties you can set in the above dialog, Validations, and rules for setting fields in your data based on values just submitted on the current page.

## Validation Settings

Validation means to inspect the value of a given field, and enforce any rule that the value should be of a certain type.

For example, if your data fields contain an integer number, such as an ID value, then validation is used to ensure that it does not contain a decimal point, and that it contains no letters, only pure digits.

To create Validation rules, click the Validations tab in Figure 1.

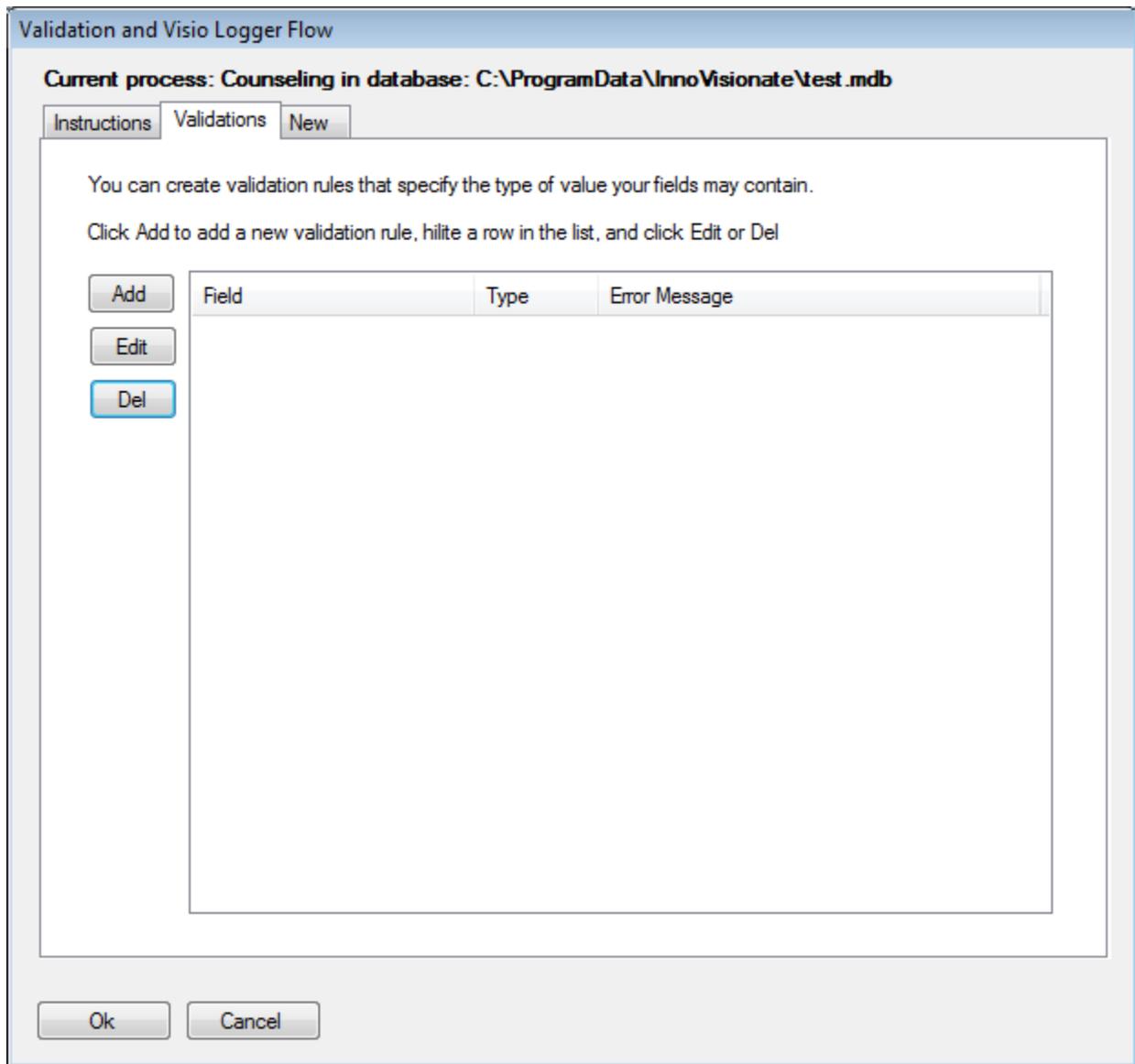


Figure 2 Validation Rules Dialog

The figure shows no validations for this database and process. To Create one, click Add, and you will see the dialog shown in Figure 3

The screenshot shows a dialog box titled "New Validation Rule". It contains two dropdown menus: "Field:" with "Student ID" selected and "must be:" with "Integer" selected. Below these is a text input field for "Message to user:" containing the text "Student ID must be an integer". Underneath the text field is the instruction "(leave blank unless the user should be notified)". At the bottom of the dialog are two buttons: "Ok" and "Cancel".

Figure 3 A New Validation Rule

The rule defined in the above figure was created by selecting the Field from the available fields in the data set, and by choosing either Integer, Decimal, or Date from the “must be:” list.

Note that the Message to user is specified for this rule. This particular configuration of Visio Logger is intended to be run in Kiosk Mode – which means that the signer can actually see the computer monitor, and therefore, can be apprised any failure of validation. When Visio Logger is *not* running in Kiosk Mode, the “Message to user:” field should be left blank, because there may be no-one at the computer to dispatch any message, and data collection would stop. When a keypad is in use – a failed validation will cause the same page to be resubmitted to the signer.

After you define each rule, they will appear in the list of rules shown in Figure 2. You can always edit or delete the rule by selecting it in that list and pressing Edit or Del to the left of the list.

### Creating Value Setting rules

Value Setting rules set the value of any field based on the value of any other field.

The “name” of each value setting rule is based on the name of the field whose value you are going to *inspect*. Therefore, there is a “group” of rules based on one field and each “group” has its own tab in the dialog shown in Figure 1.

To create a new group of rules for a field, click the “New” tab in Figure 1. When you are within that tab, you have to first select the name of the field that will be *inspected* from the list of fields.

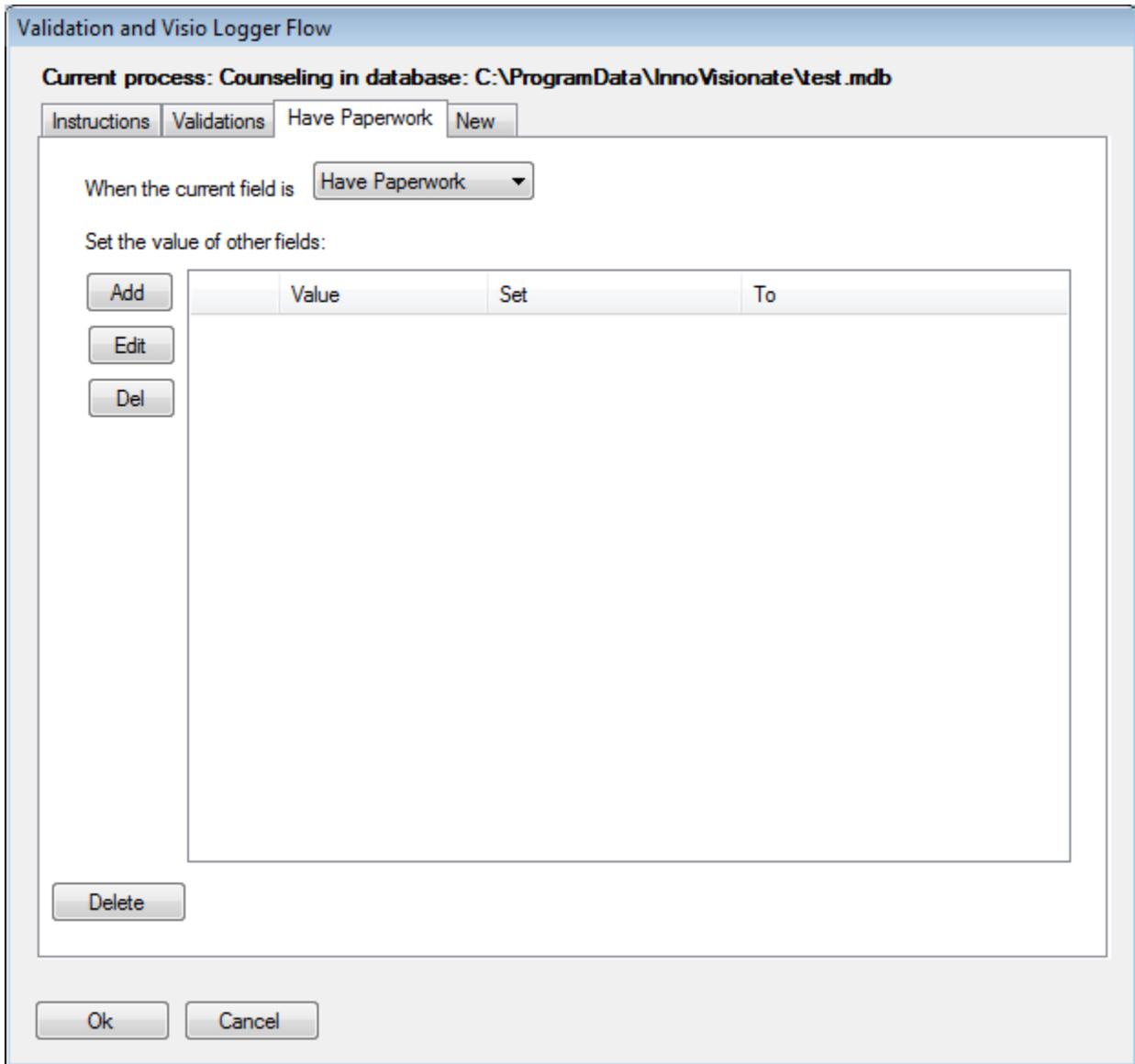


Figure 4 A new group of rules inspecting "Have Paperwork"

Note in the above, that after clicking the New tab – we then immediately chose a field from the available fields dropdown, at which point, the “New” tab was *renamed* to the chosen field – *and* an additional tab was added and *that* tab becomes the “New” tab. You can create as many tabs as there are fields in your database.

To add a new value setting rule, click Add, you will see the dialog as in Figure 5.

rule

When   IS equal   is NOT equal specify <any> to always set the following field

Set  to

(Select placeholder values here)

Figure 5 A new Value Setting Rule

Note that the “When” field is disabled – this reinforces the fact that a group of rules (the items in the list in Figure 4) are based on *one* field only, the field being *inspected*, which, in this case is “Have Paperwork”.

We fill in the dialog as follows:

- When the value of Have Paperwork IS equal to some given value, then set the value of *another* field to a specified value.

or,

- When the value of Have Paperwork IS NOT equal to some given value, then set the value of *another* field to a specified value.

The “Have Paperwork” field in the database in use happens to be a Menu data entry field. Because of this, we know that it can have one of a set number of values; there can be no misspelling of the value.

Assuming this value could be Yes or No. Then, if our database has the field, “Needs to ReSchedule”, we could create our rule as shown in Figure 6.

rule

When   IS equal   is NOT equal specify <any> to always set the following field

Set  to

(Select placeholder values here)

Figure 6 Setting data for another field

Did you know that your data fields can contain *any* number of *non*-input fields? Simple supporting fields that can hold any value at all. These are specified by turning off the selection “Is On the Entry Panel?” in the column definition.

This is what we have done with the “Needs to ReSchedule” field. Therefore, given our rule, if a student forgot to bring their paperwork – the row of data for this student will immediately note this fact, even before the student is finished with the rest of their sign in session! Reception sitting at the front desk would immediately see this if they were running Visio Logger in Viewer Mode, and could even start the rescheduling process for that student even before she knows she will have to come back.

Note our new list of rules for the “Have Paperwork” field in Figure 7.

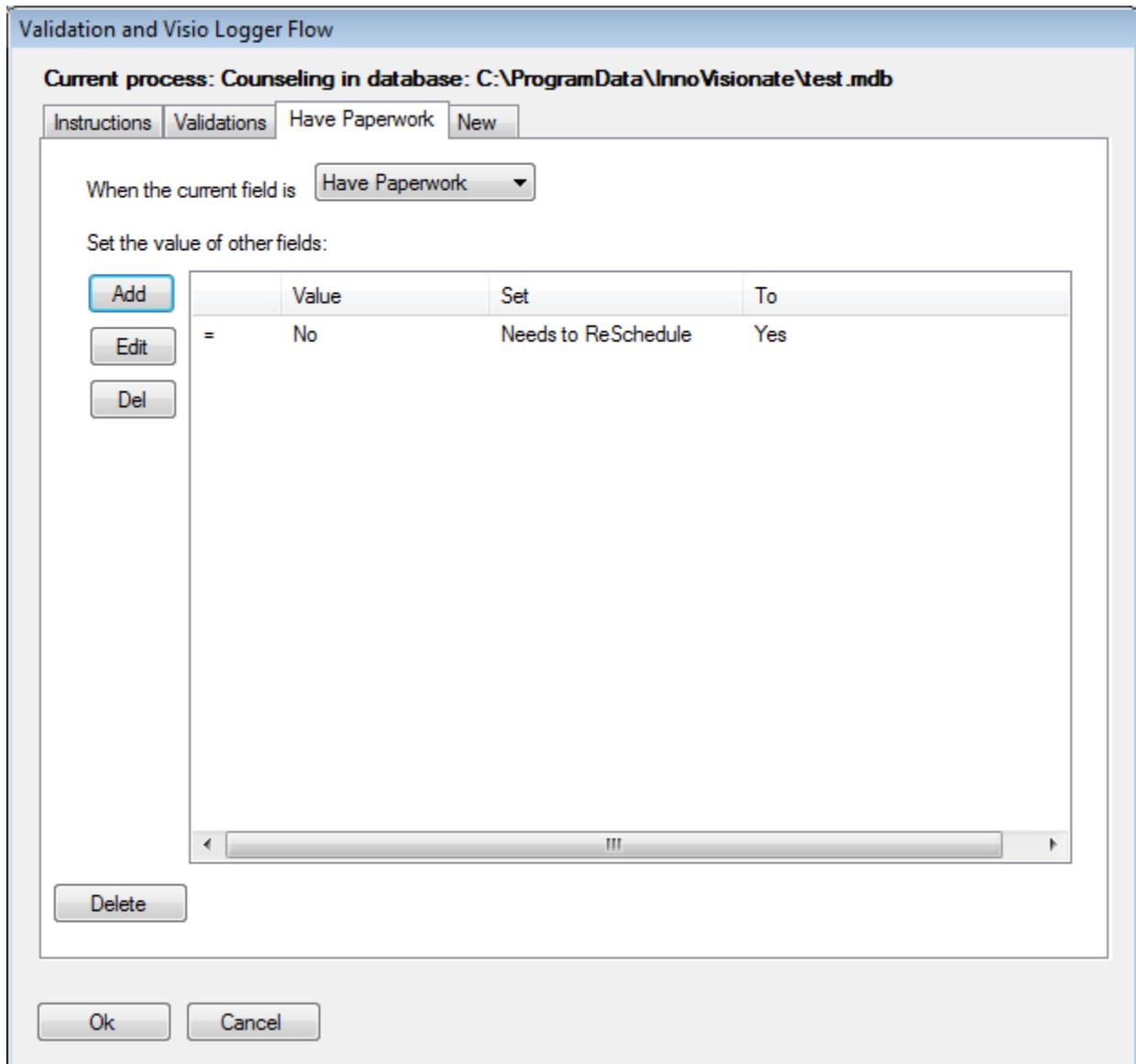


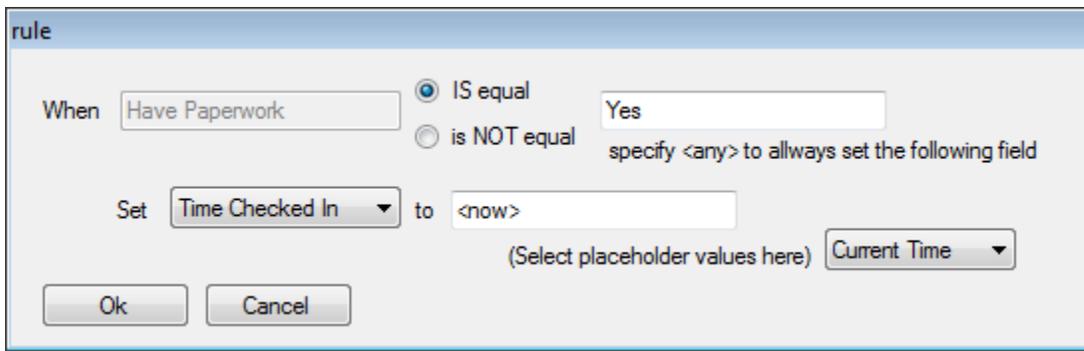
Figure 7 The list of rules

We can always edit or delete any rule by highlighting it and pressing the appropriate button to the left of the list.

We will create one more rule that uses a “placeholder” value, which is a logical name for a value to be substituted at run time.

Here we say that if the student *does* have their paperwork – we will grant them entry at the current time (we chose the “Current Time” entry in the drop down list in the bottom right).

In fact, the time the student checked in is always maintained by the system in the “Signed In” column, which is always there. However, for illustrative purposes, or as a convenience to staff perusing the Visio Logger data, we can repeat it in this field.



The image shows a dialog box titled "rule". It contains the following elements:  
- A "When" field with the text "Have Paperwork".  
- Two radio buttons: "IS equal" (selected) and "is NOT equal".  
- A text field containing "Yes" next to the "IS equal" radio button.  
- A note below the radio buttons: "specify <any> to always set the following field".  
- A "Set" field with a dropdown menu showing "Time Checked In".  
- A text field containing "<now>" next to the "Set" field.  
- A note below the text field: "(Select placeholder values here)".  
- A dropdown menu showing "Current Time".  
- "Ok" and "Cancel" buttons at the bottom.

Figure 8 A rule using a placeholder value

Both of the “target” fields in these two rules are “non-Entry” fields as mentioned before, therefore, since the Visio Logger flow through the data fields will never encounter these fields anyway, these 2 rules will not affect the flow.

However, it is important to remember that *if* these *were* entry fields (data collected from the signer in some way) – *then* setting a value for the target field (even if it is just an empty string) *WILL* cause *that* field to be skipped and the signer will never be asked to provide a value.

That is the crux of how selective “flow” through the data fields works. If you have more than one “type” of signer, or data collection situation, you will still have *all* the fields necessary in your data, however, many of them may be auto-filled with values from these rules. If desired, it might make sense to auto-fill the values with N/A to indicate the field is not applicable. Or, simply provide a blank value as you see fit.

Note also that setting target field values *is* iterative and that this can have unintended side effects. Specifically, if there is a rule that sets field 1 to the value A, when field 2 has the value B, *and* there is a rule that sets the value of field 3 to the value C when field 1 is A – *then*, whenever field 2 is given the value B – field 1 becomes A, and field 3 becomes C. This should be noted if unexpected results are occurring.

### When Signature (bitmap) fields are the target

You cannot “inspect” the value of a signature field, since it contains a bitmap. However, you *can* set the value of a signature field to blank. This way, you can collect a signature from certain types of visitors, and skip the signature for other types.

### Saving the Configuration

The configuration file you have created with these settings is stored in the local machine’s ProgramData directory. Upon exit from the properties dialog, the system will present the following question:

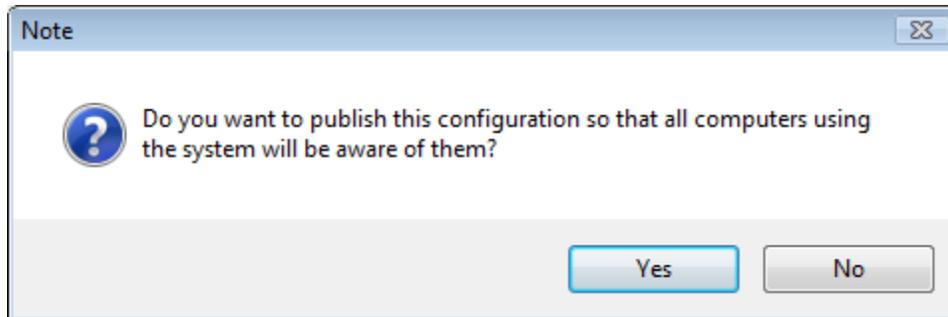


Figure 9 Prompt to save the configuration

What this option does is to store the configuration file directly in the current database in use. This way, when *another* computer on the network is using that database, and the option to use the Page Submitted Processor is selected on that computer – the processor will, at the appropriate time, check for an update to these settings and put them into effect on that other computer.

### Summary

This completes the configuration of the InnoVisioNate Page Submitted Processor.

To see it in action, be sure to check the option to the left of the name of the processor in the Tools → Toolbox dialog.