



Workflow Engine for Data Entry

Functional Specification

Version	Date	Autor	Changes
0.1	2012-12-26	Peter Pavek	Initial version
0.2	2013-02-01	Peter Pavek	Added set value block. Reformatted to have a general workflow chapter and a page formatting one.
0.3	2013-02-10	Peter Pavek	Object name property changed to “name” for all blocks.
0.4	2013-02-16	Peter Pavek	Changed property “is visible” to “is available”. Added type and unit properties to variables. This is now in synch with other documents.
0.4_1	2013-02-25	Terje Lundin	Changed the text slightly of Create Field Block to accommodate for Boolean Truth types.

Introduction

Background

Teraim develops software for mobile data collection. The key aspect of this software is its configurability and embedded data value validation.

Scope

This document describes the functionality of a component responsible for orchestrating the definition and display of data entry pages. A commonly used paradigm of workflows is used and this document describes the basic functionality and the use in this context.

Note that, while workflows and their components are described here, their syntax is not specified. Rather, this is left to the implementation specification.

Definitions

Page	Used in this document to mean the contents of the screen – both the visible parts and the parts that may be hidden, but can be scrolled to.
Variable	Data structure for storing a number of data values. A variable has an associated entry field. The graphical layout of the field depends on the type of the variable. Each variable has an “Is Available” flag. By default, it is TRUE.

Workflows

The following blocks form the required kernel for the data entry system. Each block has the possibility to have a next block. If a next block does not exist, the workflow ends.

General Workflow Blocks

A sequence of actions is implemented as a workflow. The generic concept is described in [WFE]. The specifics for this context consist of special workflow blocks and definition of execution run-time data. The workflow blocks that are required are

- General blocks
- Condition blocks
- Display blocks

The display blocks direct the system to display something. The type of display block determines what to display – a page, button on the current page, or a data entry field for a variable.

The condition blocks implement routing logic based on the value of a condition on run-time data. An example condition block would have a condition where a variable is compared to a constant, e.g., $X = 3$. If true, the block along the *then* connection is executed next. If false, the block along the *else* connection is executed next.

All blocks share the following properties:

- *Label* (text) is a name of the block. It has no functional meaning other than being displayed next to the block and remind the reader of the purpose of the block.

Start Workflow Block

This is the first block in every workflow. It’s main purpose is to carry the name of the workflow and to initialize some parameters. The block has the following properties:

- *Workflow Name* (symbol¹) is the name of this workflow. The name must be unique in the system.

¹ A symbol is a text string containing only printable, non-blank characters. The first character may not be a digit.

Set Value Block

This block assigns a value to a variable. If the variable is not already defined, this block creates the variable as well. The block has the following properties:

- *Field Label* (text) is the label of this field, when displayed on the related page.
- *Variable Name* (symbol) is the name of this variable. The name must be unique to the system. When referring to this variable, this name is used.
- *Value* (expression) is an expression giving a value to the variable. It can be a constant (number, text, or logical), a variable name, or an arithmetic or logical expression of these.

Add Rule Block

This block defines a rule to be added to a variable or button (see below.) Rules are used to validate entered values, control the visibility of objects, set parameters, etc. This functionality is described in [Rules]. A sequence of these blocks can add several rules to one entity. The block has the following properties:

- *Object Name* (symbol) is the name of the variable or button to receive this rule. The object must be defined using a Display Variable block prior to this block.
- *Condition* (logical expression) see [Rules], section Rule Conditions.
- *Action* (action sequence) see [Rules], section Rule Actions.
- *Error Message* (text) is the text to be shown on the screen in case the condition is executed and is evaluated to FALSE. Note that this is only relevant in case that this is a validation rule, i.e., the action states that a value is valid.

Condition Block

This is a condition block that directs the continuation of the execution depending on the value of the condition expression. The block has the following properties:

- *Condition* (logical expression) that may contain constants, variables, and logical combinations of these. This is analogous to the description given in [Rules], section Rule Conditions.

The block has two “next block” alternatives – one is used when the condition evaluates to TRUE (this is the *then* alternative) and one when the condition evaluates to FALSE (this is the *else* alternative.)

Display Management

The goal of this function is the management of page and variable display. The management consists of issuing commands for displaying a page and, on the page, display a sequence of objects, e.g., variable entry fields and buttons. The display of the objects may follow a single, linear path. However, there may be conditions associated with the display of variables as well as formatting instructions.

A typical use case would be when a new page is displayed containing a number of fields, where values for the associated variables can be filled in.

Define Page Block

This block defines a new page and directs the workflow engine to display it. The block has the following specific properties.

- *Name* (symbol) is the name of the page to display. The name must be unique in the system. When referring to this page, this name is used.
- *Page Label* (text) is the text that appears at the top of the page. If left empty, the page will have no header.
- *Is Available* (logical) is a flag that controls the visibility of the page. As long as the flag is true, the workflow engine will display the page.
- *Page Format* (style or format type) is a description giving hints as to how the rendering engine should draw the objects on the page. [optional extension, nice to have]

Create Field Block

This block defines a field for editing a variable value. If the associated variable is not already defined, this block creates the variable as well. The block has the following properties:

- *Field Label* (text) is the label of this field, when displayed on the related page.
- *Variable Name* (symbol) is the name of this variable. The name must be unique to the system. When referring to this variable, this name is used.
- *Variable Type* (symbol) is the name of one of the predefined variable types. Examples are integer, number, text, truth-value, percentage, date, and currency. The value of this property directs the selection of the editor. A numeric type will bring up a number pad. A text type will bring up an alphanumeric keyboard. A truth-value type will bring up Radio Buttons for making the selection.
- *Unit* (text) is the unit used for measuring the variable value. Examples are “m” and “dm”. Empty strings are allowed.
- *Page Name* (symbol) name of the page where the variable entry field should appear.
- *Display Purpose* (text) controls the display field. May have values *display* or *edit*. The value *display* indicates that the variable (and its value) should be shown on the page, but it may not be edited. The value *edit* indicates that the variable’s value may be edited.
- *Is Available* (logical) is a flag that controls the “visibility” of the variable. As long as the flag is true, the workflow engine will display the entry field on the referenced page. Note that this flag is global, in that all fields referencing this variable are affected.
- *Show Previous Value* (logical) is a flag controlling whether to show the value from the previous sampling or not.

XML Representation & example:

```
<block_create_field>
  <label>Create Field</label>
  <varname>A</varname>
  <vartype>numeric</vartype>
  <pagename>?</pagename>
  <purpose>edit</purpose>
```

</block_create_field>

Create Check Box Block

This block defines a new check box button on a page. The button controls (and displays) the value of an associated variable. The button may have rules associated with it, that control the visibility of the button. The block has the following properties:

- *Name* (symbol) is the name of the button. The name must be unique to the system.
- *Button Label* (text) is the text to appear as the label of the button.
- *Variable* (symbol) is the name of the variable associated with this check box.
- *On Value* (text) is the value to be assigned to the variable when the button is checked.
- *Off Value* (text) is the value to be assigned to the variable when the button is checked.
- *Is Available* (logical) is a flag that controls the visibility of the button. As long as the flag is true, the workflow engine will display the button on the referenced page.
- *Page Name* (symbol) name of the page where the button should appear.

Create Drop-down List Block

This block defines a new drop-down list button on a page. The button controls (and displays) the value of a logical variable. The button may have rules associated with it, that control the visibility of the button. The block has the following properties:

- *Name* (symbol) is the name of the button. The name must be unique to the system.
- *Button Label* (text) is the text to appear as the label of the button.
- *Variable* (symbol) is the name of the variable associated with this check box.
- *Values List* (CSV list) is a comma-separated list of values that are to appear in the drop-down.
- *Is Available* (logical) is a flag that controls the visibility of the button. As long as the flag is true, the workflow engine will display the button on the referenced page.
- *Page Name* (symbol) name of the page where the button should appear.

Create Action Button Block

This block defines a new button on a page. The button has an action that is performed when the button is pressed. Currently, the action is to launch a workflow. The button may have rules associated with it, that control the visibility of the button. The block has the following properties:

- *Name* (symbol) is the name of the button. The name must be unique to the system.
- *Button Label* (text) is the text to appear as the label of the button.
- *Action* (symbol) name of the workflow (or other action) to execute when the button is pressed.
- *Is Available* (logical) is a flag that controls the visibility of the button. As long as the flag is true, the workflow engine will display the button on the referenced page.
- *Page Name* (symbol) name of the page where the button should appear.

Set Appearance Block

This block sets the appearance of objects on a page. Initially, this block affects the availability and coloring. The block has the following properties:

- *ObjectName* (symbol) is the name of the object (button or data entry field.) The name must be unique to the system.
- *Action* (?) name of the workflow (or other action) to execute when the block is executed. Example actions are “set *NN* as available” and “set *NN* {background | foreground | text} to red.”

Examples

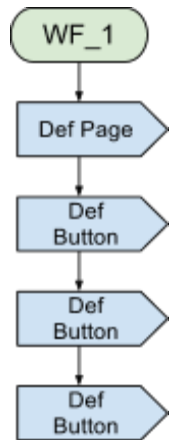
This section contains a number of example workflows. The workflows are depicted graphically along with the resulting pages. The appearance and layout is not at all part of the functional requirement and is intended for illustrative purposes only.

Example 1 – Index Page

This example shows a workflow for creating a page with buttons leading to other pages. The page has a header text and three buttons. The last button is disabled, as its *Is Available* property is false.



The workflow that created the page is as follows. Note that the graphical appearance is for illustration purposes only.



The properties of the blocks are as follows:

Block 1: Start Workflow

Label: WF_1

Name: WF_1

Block 2: Define Page

Label: Def Page

Name: Page_1

Page Label: Available Sections

Is Available: true

Page Format: left-aligned

Block 3: Create Action Button

Label: Def Button

Name: Button_1

Button Label: Section 1

Action: WF_Section_1

Is Available: true

Page Name: Page_1

Block 4: Create Action Button

Label: Def Button

Name: Button_2

Button Label: Section 2

Action: WF_Section_2

Is Available: true

Page Name: Page_1

Block 5: Create Action Button

Label: Def Button

Name: Button_3

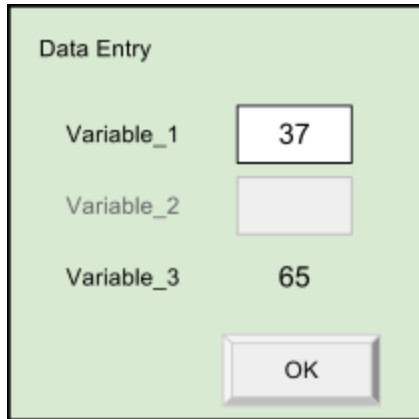
Button Label: Section 3

Action: WF_Section_3

Is Available: false
Page Name: Page_1

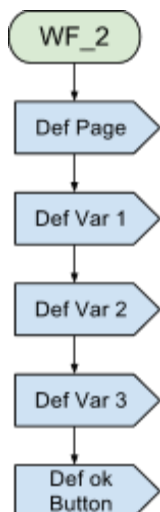
Example 2 – Data Entry Page

This example shows a workflow for creating a page for data entry.



There are three variables – one that can be edited, one that is “not visible”, and one that Is Available but can’t be edited. In general, the variables that are editable can be selected and a suitable keyboard appears. In the figure above, the first variable is selected.

The following workflow could have generated the above page.



The properties of the blocks are as follows:

Block 1: Start Workflow

Label: WF_2
Name: WF_2

Block 2: Define Page

Label: Def Page
Name: Page_2

Page Label: Data Entry
Is Available: true
Page Format: left-aligned

Block 3: Create Field

Label: Def Var 1
Field Label: Variable_1
Variable Name: Variable_1
Page Name: Page_2
Is Available: true
Show Previous Value: true

Block 4: Create Field

Label: Def Var 2
Field Label: Variable_2
Variable Name: Variable_2
Page Name: Page_2
Is Available: false
Show Previous Value: true

Block 5: Create Field

Label: Def Var 3
Field Label: Variable_3
Variable Name: Variable_3
Page Name: Page_2
Is Available: true
Show Previous Value: true

Block 6: Create Action Button

Label: Def Button
Name: Button_1
Button Label: Section 1
Action: WF_2_ok
Is Available: true
Page Name: Page_1

Example 3 – Data Entry Page with Rules

This example shows a data entry page with two variables, both containing validation rules and one of them containing also a rule controlling the visibility. The variables in this example do not have old (previously collected) values. Thus, there are not shown in the entry field.

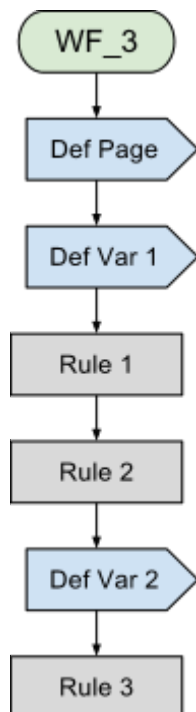
Controlled Variables

Variable31

Variable32

None of the variable fields is selected in the above page and both are editable.

The following workflow could have generated the above page.



The properties of the blocks are as follows:

Block 1: Start Workflow

Label: WF_3

Name: WF_3

Block 2: Define Page

Label: Def Page

Name: Page_3

Page Label: Controlled Variables

Is Available: true

Page Format: left-aligned

Block 3: Create Field

Label: Def Var 1

Field Label: Variable31

Variable Name: Variable31
Page Name: Page_3
Is Available: true
Show Previous Value: true

Block 4: Add Rule

Label: Rule 1
Object Name: Variable31
Condition: $\text{Variable31} > 0$ and $\text{Variable 31} < 10$
Action: value is valid
Error Message: “The value is outside the allowed range, 0–31.”

Block 5: Create Field

Label: Def Var 2
Field Label: Variable32
Variable Name: Variable32
Page Name: Page_2
Is Available: false
Show Previous Value: true

References

- [WFE] Generic Workflow Engine, 2012, Pavek, P.
[Rules] Rules for Validation, 2012, Pavek, P.