



WORKFLOW BUILDER™ FOR MICROSOFT® ACCESS™

Application Guide
Version 06.05.2008

This document is copyright © 2007-2008 OpenGate Software. The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

Workflow Builder is a trademark of OpenGate Software Inc.

Microsoft and the Office logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.



TABLE OF CONTENTS

1	GENERAL CONCEPTS.....	3
2	CREATING A WORKFLOW	3
2.1	GENERAL WORKFLOW SETTINGS.....	3
2.2	DEFINING WORKFLOW STEPS	4
2.3	WORKING WITH STRING REPLACEMENTS.....	7
2.4	PERFORMING MASS FILE IMPORTS	9
2.5	RECORD UPDATE NOTIFICATIONS	9
2.6	VALIDATING WORKFLOWS AND WORKFLOW STEPS	10
2.7	VISUALIZING WORKFLOWS.....	11
3	RUNNING A WORKFLOW	12
3.1	RUNNING FROM THE WORKFLOW ADMIN FORM.....	12
3.2	STOPPING A WORKFLOW	13
3.3	RESTARTING A WORKFLOW	13
3.4	RUNNING A WORKFLOW FROM CODE.....	13
3.4.1	<i>Workflow Console.....</i>	<i>14</i>
3.5	SCHEDULING WORKFLOWS TO RUN AUTOMATICALLY	14
4	EXPORTING/IMPORTING WORKFLOWS	16
4.1	EXPORTING A WORKFLOW	16
4.2	IMPORTING A WORKFLOW	16
5	WIZARD FORMS	16
5.1	CONFIGURING A WIZARD PROCESS.....	17
5.1.1	<i>General Wizard Setup.....</i>	<i>17</i>
5.1.2	<i>Configuring Wizard Steps.....</i>	<i>18</i>
5.2	REQUIRE VALIDATION (OPTIONAL)	19
5.3	SKIPPING STEPS (OPTIONAL).....	19
5.4	BRANCHING TO ANOTHER WIZARD (OPTIONAL).....	20
5.5	LAUNCHING A WIZARD.....	20
5.6	EXAMPLE WIZARDS.....	21
6	TROUBLESHOOTING WORKFLOWS.....	21
6.1	WORKFLOW LOGS	21
6.1.1	<i>Sequence Identifiers.....</i>	<i>21</i>
6.2	WORKFLOW SETUP ISSUES.....	22
7	RELEASE HISTORY	22



1 General Concepts

Workflow Builder™ allows users to create an unlimited number of workflows and process steps for each workflow. Each workflow contains one or more process steps, each of which attempts to complete a specified operation. Each process step also has a Success Action and Failure Action that will be attempted based on the outcome of the primary step.

2 Creating a Workflow

2.1 GENERAL WORKFLOW SETTINGS

Each workflow you establish has a set of general settings that define how Workflow Builder processes your workflow. The general workflow settings are configured in the Workflow Administration screen, shown below.

The screenshot shows the 'Workflow Administration' screen in Workflow Builder. It features a sidebar with navigation options: Main Menu, Test, Workflows (selected), Reports, Configure, Exit, and OpenGate Software. The main area displays a table of workflows and configuration options.

Workflow Name	Status	Frequency	Last Run	Next Run	Run on Weekends?	Validation Status
Long Workflow Examp	Active	Manual	3/17/2008 12:59:01 AM		<input type="checkbox"/>	Valid
Short Workflow Examp	Active	Manual	3/17/2008 12:59:01 AM		<input type="checkbox"/>	Valid
Failure Example	Active	Manual	3/17/2008 12:59:01 AM		<input type="checkbox"/>	Valid
*	Active	Manual			<input type="checkbox"/>	

Below the table, there are 'Advanced Options for This Workflow' including a dropdown for 'On Full Failure' (set to 'Nothing') and a text field for 'Failure Email To:'. There are also checkboxes for 'Show Workflow Console when running' (checked) and several action buttons: 'Run Selected', 'Visualizer', 'Validate Selected', 'Disable All', 'Manage Selected', and 'Delete Selected'.

- Workflow Name** Identifies the common name for the workflow.
- Status** Set to "Active" to allow users to run this workflow. Setting the field to "Disabled" will prevent users from running the workflow.
- Frequency** When the AutoRun feature is active, the Frequency selected will dictate how often Workflow Builder runs a workflow.



- Next Run** When AutoRun is active, the Next Run field determines when the next automatically scheduled execution of workflow will occur.
- Run on Weekends** If unchecked, the workflow will not run on Saturdays and Sundays if AutoRun is active.
- On Full Failure** You may specify whether what the selected workflow will do if both the primary action and failure actions cannot be completed. Options are:
- **Nothing** – Workflow Builder will simply end when a full failure condition is met.
 - **Email** – Workflow Builder will send a failure email message to the designated recipients when a full failure condition is met.
 - **Alert Dialog** – Workflow Builder will display a dialog box with a failure message when a full failure condition is met.
 - **Alert Dialog and Email** – Both an email and dialog will be created.
 -
- Failure Email** If you choose **Email** or **Alert Dialog and Email** in the On Full Failure field, you must supply one or more email addresses, separated by semi-colons.
- Show Workflow Console** This checkbox will apply to all workflows, and determines whether Workflow Builder will display the detailed processing console with extensive detail about processing each step.

2.2 DEFINING WORKFLOW STEPS

For each workflow you create, you may define an unlimited number of processing steps. Each processing step may also have a success and failure action.

Defined below are the available step actions packaged with Workflow Builder. Note that some actions are not available for the Success and Failure action paths. For example, if the workflow fails, Export to File is not an available failure path to choose from.

Step Action	Description
Alert Dialog	Displays a standard dialog box with the message text you specify. Note that this action will halt the workflow until a user intervenes by clicking the "OK" button.
Alert Popup	Displays a popup dialog that will fade away after a predefined number of seconds. Note that this action will not halt the workflow.



Step Action	Description
Clear Table	Clears (deletes) all records in a specified table. Similar to the Delete Record command, but no SQL criteria is needed.
Continue	In cases where a step fails, setting the Failure Action to "Continue" will cause Workflow Builder to proceed to the next step in the workflow.
Copy File	Copies a specified file to another location.
Copy Table	Copies a specified table to a new name you define.
Delete File	Deletes a specified file.
Delete Record	Deletes one or more records that meet your specified SQL Where criteria.
Delete Table	Deletes a table contained in the local database.
Email Report	Emails a specified report in PDF format.
Export to Excel	Exports data in a specified table or query to Excel.
Export to Text (delimited)	Exports data in a specified table or query to a delimited text file. You must define the export specification in the local Access database for this action to function correctly.
Export to Text (fixed width)	Exports data in a specified table or query to a fixed width text file. You must define the export specification in the local Access database for this action to function correctly.
Import Excel	Imports an Excel file into the table of your choice. Note that if the destination table already exists, and the names and number of fields in the destination table do not match the names and number of fields in the Excel file, Microsoft Access will not be able to import the file.
Import File (fixed width)	Imports a fixed width text file into the table of your choice. You must define the import specification in the local Access database for this action to function correctly.
Import Text (delimited)	Imports a delimited text file into the table of your choice. Note that you must define the import specification in the local Access database for this action to function correctly.



Step Action	Description
Log Recordcount	Obtains a count of the records contained in a table you specify. This is helpful, as an example, where you would like to identify the number of records imported, or the number of records before and after you run an action query against a table.
Move File	Moves a specified file to another location.
Open File	Opens a specified file using the registered Windows application. For example, you may wish to open a PDF document or execute a batch file during a workflow process.
Record Update Notification	Searches for new, changed, or new and changed records in a specified table, and generate a notification if any are located.
Rename Table	Renames a table in the database. Note that this function will only work for local tables stored in your database. Linked tables cannot be renamed with this method.
Run Code	<p>Attempts to execute a VBA function you specify. To pass in variables to the function, be sure to follow this convention:</p> <pre>YOURFUNCTIONNAME("TEXTVARIABLE",0)</pre> <p>Where in the example above, any text variables are placed in quotes, and numeric variables are not in quotes. You do not need to precede your function names with an '=' sign.</p> <p>We strongly recommend your custom functions return a Boolean True/False value to indicate whether the function completed successfully (True) or failed (False). If a non-Boolean value is returned, Workflow Builder assumes your function succeeded.</p>
Run Macro	Attempts to execute a macro in your database. Note that if your macro fails, the user will typically be prompted to dismiss an error message dialog, which will pause your workflow until they do so. To ensure your workflows process even without user input, we recommend you replace macros with VBA functions.
Run Query	Runs a specified query in the local database.
Run Report (preview)	Opens a specified report in preview mode.



Step Action	Description
Run Report (print)	Opens a specified report and automatically prints the report using the default printer for the local machine.
Simple Email	Sends an email message to the recipients you specify using Microsoft Outlook.
Stop	Halts the workflow entirely.
Update Record	Updates the value of a field for a record or records in a table that meet your specified SQL Where criteria. For example, you may wish to execute a report, and then update a Boolean field in a table when the report has been run (to prevent those records from appearing in the record again).
Wait and Retry	Attempts to retry the primary action after a specified period of time for the number of tries you define. For example, if the step indicates a file should be imported, and the file does not exist, Workflow Builder be configured to retry the import step n-times until the file is present and can be imported. This step action is only available in the Failure Action dropdown.

2.3 WORKING WITH STRING REPLACEMENTS

In some instances, you may wish to include dynamic values in your workflow operations that cannot be identified at the time the workflow is created. For example, perhaps your application needs to import a text file from a known directory each day, however the text file will have a different file name each day that includes the Julian date.¹ In this case, the file to import on January 18 of each year may be:

C:\FileName18.txt

And the file to import on March 18th would be:

C:\FileName77.txt

To include string replacement in a workflow step, you include one of the following keywords in your step definition:

Keyword	Purpose
---------	---------

¹ The number of days elapsed since the first day of the current year. For example, January 10th is the Julian day 10, an February 1st is Julian day 32.



Keyword	Purpose
[Day]	Replaces the keyword with the current day.
[Month]	Replaces the keyword with the current month.
[Year]	Replaces the keyword with the current year.
[Minute]	Replaces the keyword with the current minute.
[Second]	Replaces the keyword with the current second.
[Julian]	Replaces the keyword with the current Julian day.
[Username]	Replaces the keyword with the Windows username for the individual currently logged in.
[Machine]	Replaces the keyword with the machine on which Workflow Builder is running.
[DBPath]	Replaces the keyword with the current database file's path (directory).
[FileOpen]	Prompts the user to select a file using the Windows File Open dialog. If no file is chosen, a blank value is returned.
[BrowseFolder]	Prompts the user to select a folder using the Windows Browse Folder dialog. If no folder is chosen, a blank value is returned.
[Param1]	Replaces the tag with the optional varParam1 value passed in to the fRunWorkflow function when starting the workflow. If no varParam1 value is present, a null value will be returned.
[Param2]	Replaces the tag with the optional varParam2 value passed in to the fRunWorkflow function when starting the workflow. If no varParam2 value is present, a null value will be returned.
[Param3]	Replaces the tag with the optional varParam3 value passed in to the fRunWorkflow function when starting the workflow. If no varParam3 value is present, a null value will be returned.

As a final example, if you define a step of your workflow to import an Excel file, you might define the destination table as follows:

```
tbl[Month][Year][Second][Username]import
```

Which would, at the time the workflow imports the Excel file, name the table as follows:



Tbl02200859JSIMPSONimport

Where the workflow ran in February 2008 on the 59th second of the current minute for user JSIMPSON.

2.4 PERFORMING MASS FILE IMPORTS

Workflow Builder supports the ability to include a wildcard character in the path defined in the “File to Import” field. Doing so will cause Workflow Builder to attempt to import any file matching the specified path string in the directory. Files that are imported are moved to a new subdirectory “Done.” Files that cannot be imported are moved to a new subdirectory “Error.”

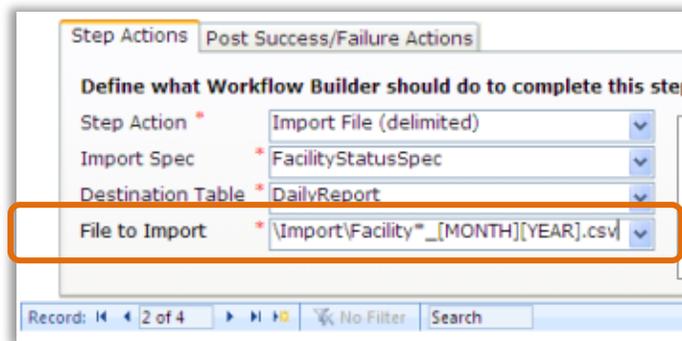


Figure 1: Example Mass Import Path

In the example shown above, any CSV files in the directory C:\Import\ with a file name similar to “Facility*_” followed by the current month and year will be imported by Workflow Builder. For example, the following files would be imported in August 2008:
 C:\Import\Facility99_082008.csv
 C:\Import\FacilityABC_082008.csv

2.5 RECORD UPDATE NOTIFICATIONS

Workflow Builder supports the ability to identify new records and/or modified records contained in a table you specify. The Step Action name is “Record Update Notification.”



Action Details for: Find Record Updates

Step Actions | Post Success/Failure Actions

Define what Workflow Builder should do to complete this step

Step Action *	Record Update Notification	▼
Table to watch *	tblOrder	▼
Date Update Field *	Date-Updated	▼
Alert on *	Both	▼
Notification mode *	Short Email brandon@opengatesw.net	▼

Figure 2: Example Record Update Notification

The record update notification process works by searching the table you specify for any records where the Date Update field in your table is greater than the date contained in a new field "WFB-Last-Update" in your table. If the field "WFB-Last-Update" does not exist in your table, Workflow Builder will ask if you would like to have it automatically added to your table (we suggest you allow Workflow Builder to do so). The Access forms in your database that update information in your specified table will need to have code added to update your date update field when a change occurs in one of two ways:

- A) By adding GotFocus and LostFocus events that detect when a value change has occurred in the field, and change the your date update field. Refer to the form "frmSilentWorkflowExample_subform" in Workflow Builder to see this method of implementation.
- B) Or by adding an OnChange event to each field on the form with the following code:

Me.Date_Updated.Value = Now()

Note that you should not perform any updates to the "WFB-Last-Update" field, and you may name your own date updated field to any name you wish. If you choose to be notified of new records and/or changed records via email, select "Short Email" in the Notification Mode, and enter the email address(es) you want the notification email to be sent to directly after the text "Short Email" as shown in the example above.

2.6 VALIDATING WORKFLOWS AND WORKFLOW STEPS

To ensure your workflows process correctly, Workflow Builder will require each step within a workflow to be completely populated based on the specific required parameters for a given step. You may attempt to validate a workflow and all steps at once, or validate each workflow



step individually. To do so, click twice on the Validation Status field for the workflow or step. If the result is Invalid, you must ensure the required fields are completed for the Step Action, Success Action, and Failure Action.

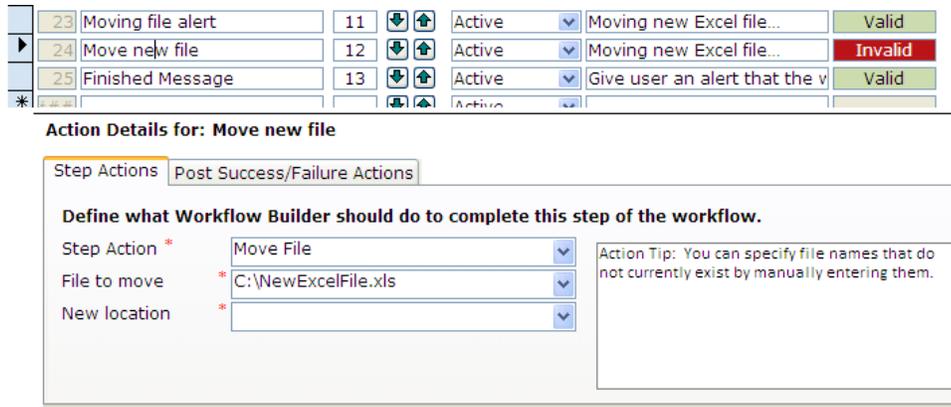


Figure 3: Example Step with missing values

2.7 VISUALIZING WORKFLOWS

In some cases it may helpful to visualize a workflow and it's steps in a single flow. To do so, open the Workflow Administration form, select a workflow, then select the **Visualizer** button.

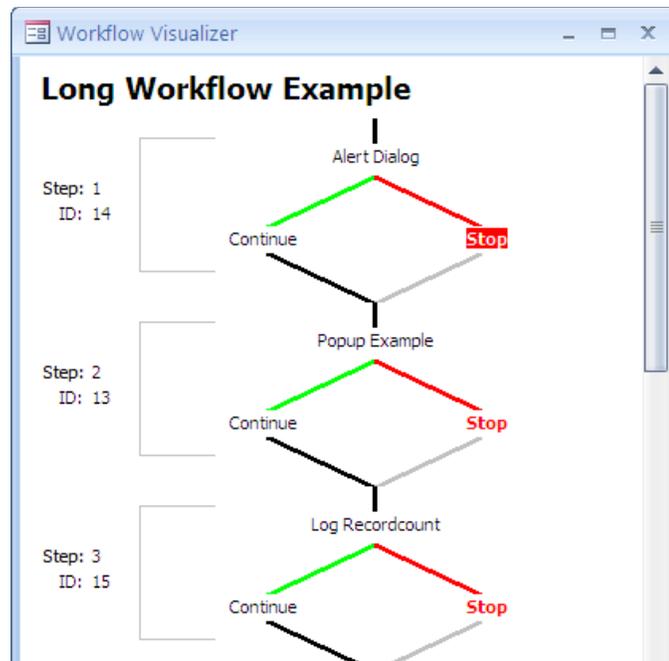


Figure 4: Workflow Visualizer



3 Running a Workflow

3.1 RUNNING FROM THE WORKFLOW ADMIN FORM

To run a specific workflow, open the Workflow Administration form, select the workflow, then click on the "Run Selected" button.

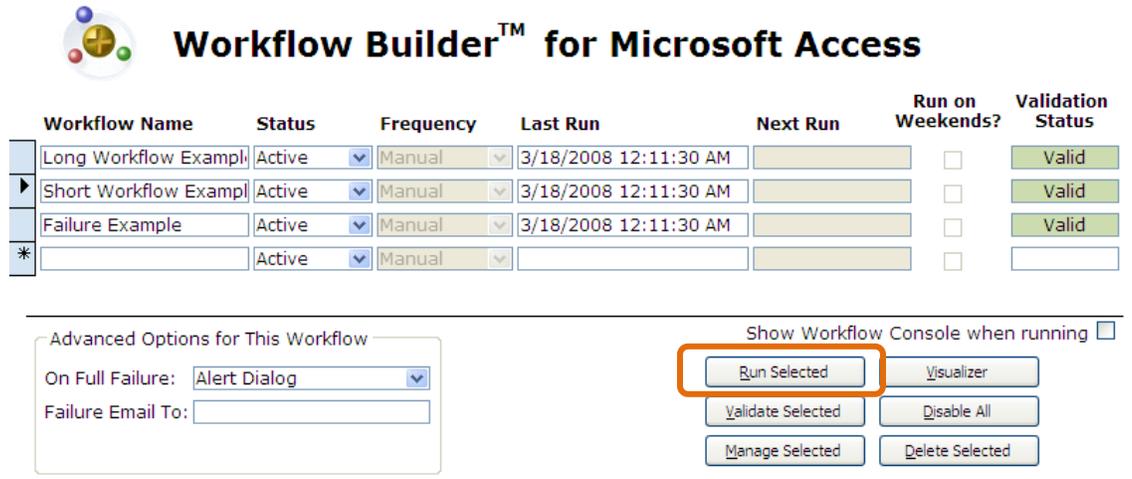


Figure 5: Workflow Administration screen

You may optionally select the "Show Workflow Console when running," which will display a detailed view of the workflow as it is processing. Workflow Builder will also open the Workflow Monitor window

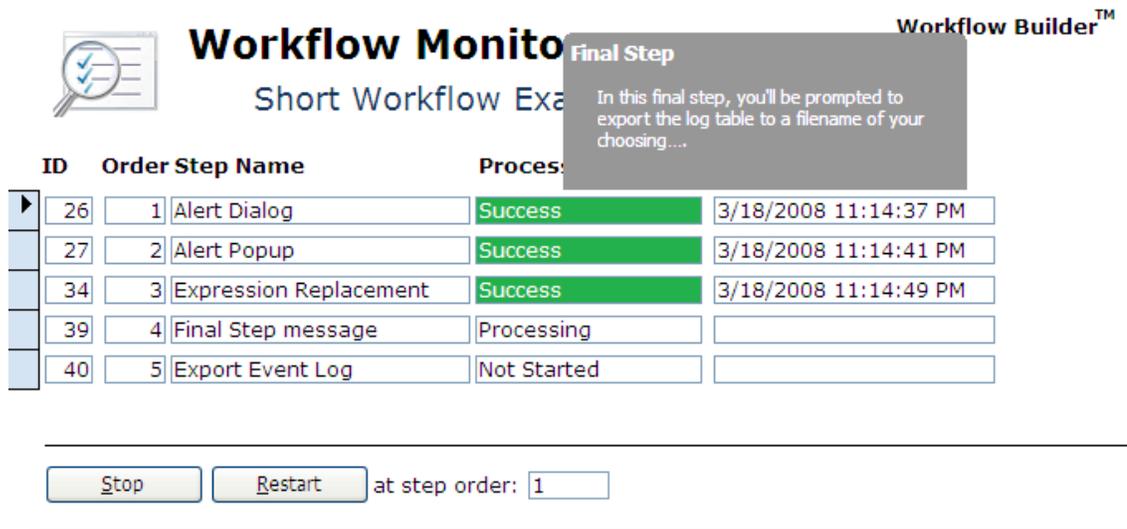


Figure 6: Workflow Monitor screen

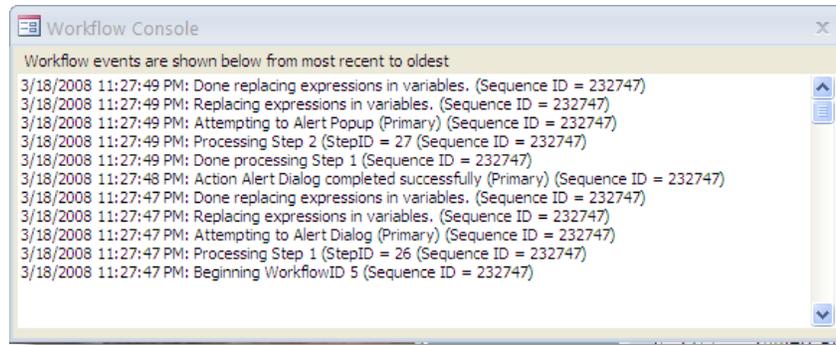


Figure 7: Workflow Console

3.2 STOPPING A WORKFLOW

Once a workflow has started processing, you may have a need to halt the workflow prior to completion. Pressing the **Stop** button will halt your workflow at the earliest possible moment, typically between two steps.

3.3 RESTARTING A WORKFLOW

Should you need to restart a workflow, you may do so at the beginning, or specify the step Workflow Builder should resume processing. To do so, enter the step to resume in the "at step order" text box, and click the **Restart** button.

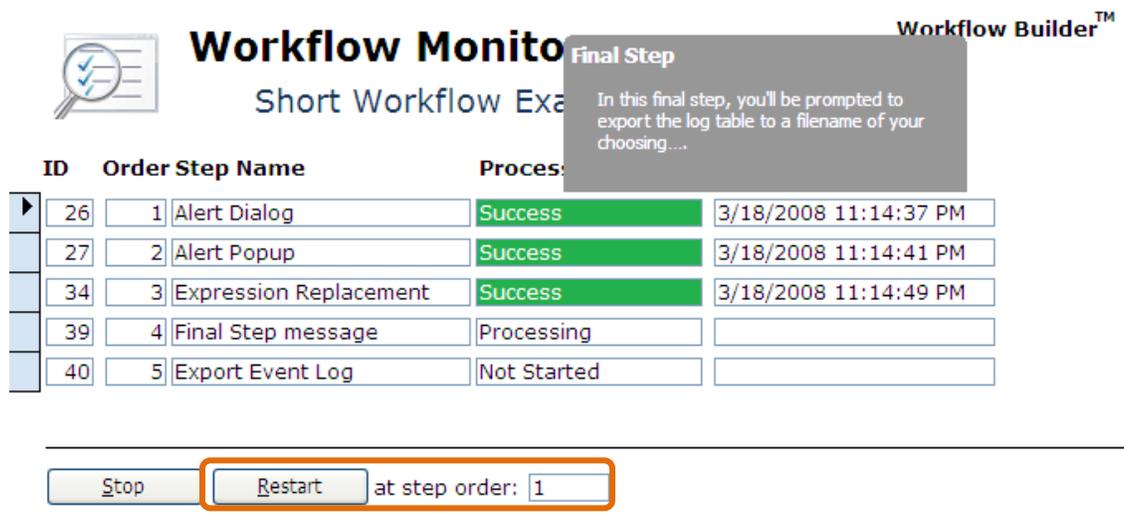


Figure 8: Restart Option

3.4 RUNNING A WORKFLOW FROM CODE

You may choose to launch a workflow directly from your VB code if needed by using the following command:



fRunWorkflow *IngWorkflowID*, *blnValidateFirst*, *iStartFromStep*, *varParam1*, *varParam2*, *varParam3*

IngWorkflowID – The WorkflowID for the workflow you want to run. This can be found in the table tblWorkflow.

blnValidateFirst – Optional. Determines if Workflow Builder will attempt to validate the setup for all steps prior to running the workflow. If this is set to False, only a cursory validation will occur.

iStartFromStep – Optional. Pass in the step number of the workflow that you want to begin with. The value will default to Step 1 if no value is provided.

varParam1 – Optional. Passes a variant parameter throughout the workflow process that is replaced any time the string [Param1] is found in the details of any workflow step. This is useful if you need to pass in a value such as a customer name or identifier to use in an action triggered in a workflow, such as an Update Record action.

varParam2 – Optional. Passes a variant parameter throughout the workflow process that is replaced any time the string [Param2] is found in the details of any workflow step.

varParam3 – Optional. Passes a variant parameter throughout the workflow process that is replaced any time the string [Param3] is found in the details of any workflow step.

3.4.1 WORKFLOW CONSOLE

To view the Workflow Console while your workflow is processing, simply set the global variable *blnShowConsole* equal to 'True' prior to calling the fRunWorkflow function.

For example, the following statements in your VB code:

```
blnShowConsole = True
```

```
fRunWorkflow 88,False,5
```

Will run workflow 88 without validating each step prior to running, begin at step 5 of the workflow, and also show the console window while processing.

3.5 SCHEDULING WORKFLOWS TO RUN AUTOMATICALLY

You can automatically execute workflows on a scheduled basis by selecting a frequency value in the "Frequency" field as shown below.



ID	Workflow Name	Status	Frequency	Last Run	Next Run	Run on Weekends?
10	Daily Report	Active	Manual	4/27/2008 11:53:03 PM		<input checked="" type="checkbox"/>
11	New Order	Active	Manual	4/27/2008 11:53:03 PM		<input checked="" type="checkbox"/>
12	AutoRun Example	Active	Every 30 min	4/27/2008 11:53:03 PM	4/26/2008 11:50:40 PM	<input checked="" type="checkbox"/>

Figure 9: Workflow Frequency Setup

Any workflow with a frequency of “Manual,” or where the field “Next Run” is empty, will not run automatically. You must also activate the “AutoRun” feature by selecting one of the options in the Workflow Management form as shown below:

Figure 10: AutoRun Configuration

Activating AutoRun will cause the AutoRun form to open every time your database opens. This form allows you to control whether your workflows run, or whether you want to halt or delay the AutoRun process. The AutoRun form will pause 60 seconds from the time your database is opening to allow you to stop your workflows from running automatically.

Figure 11: AutoRun Form

If no workflows are scheduled to run when the AutoRun form begins to process, Workflow Builder will wait 15 minutes and then check again for any workflows that need to be executed.

Note
 You may minimize the AutoRun form without affecting its behavior if you wish to work in your database. If a workflow



begins to process, however, it will interrupt your ability to work in your database.

If a workflow's Next Run time has passed because your database was not open, or there was an error that halted processing of a previous workflow, the workflow will run whenever Workflow Builder is able to successfully begin processing again.

Important!

Workflow Builder can only run workflows automatically when the database is open. If your database is closed, Workflow Builder will not be able to run your scheduled workflows until the next time the database is opened.

4 Exporting/Importing Workflows

Workflow Builder allows you to export and import workflows in Excel format. This allows you to transfer workflows between machines, or backup a workflow definition.

4.1 EXPORTING A WORKFLOW

To export a workflow, select the "Export Selected" button in the Workflow Administration screen. Exporting a workflow will cause two Excel files to be deposited in the directory of your choice. The Excel files will be named:

Workflow Name # Definition.XLS
Workflow Name # Steps.XLS

Where # is the Workflow ID assigned to the workflow. The Definition file contains the basic parameters of the Workflow, and the Steps file contains the specific actions and order of the steps within the workflow.

4.2 IMPORTING A WORKFLOW

To import a workflow, select the "Import" button in the Workflow Administration screen. You will be prompted to select the Excel file that ends with "...Definition.XLS." The imported workflow will be assigned a new Workflow ID.

5 Wizard Forms

Workflow Builder 2.0 includes a powerful new feature that lets you create a wizard process for your users without necessarily altering your existing forms. You define a set of steps you want your users to follow, including the forms that should be displayed, and any help text you want to provide to them during the process. Workflow Builder will then display a



wizard form over your existing forms that walks the user through the process you have defined.

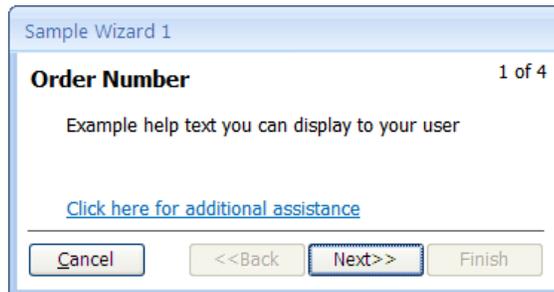


Figure 12: Wizard Overlay Form

The wizard overlay form, shown above, is configured by you during the setup process. You call the wizard process from a single line of VB code that will be provided in this section.

5.1 CONFIGURING A WIZARD PROCESS

5.1.1 GENERAL WIZARD SETUP

Creating and managing wizards is accomplished in the Wizard Administration form (frmWizardAdmin).

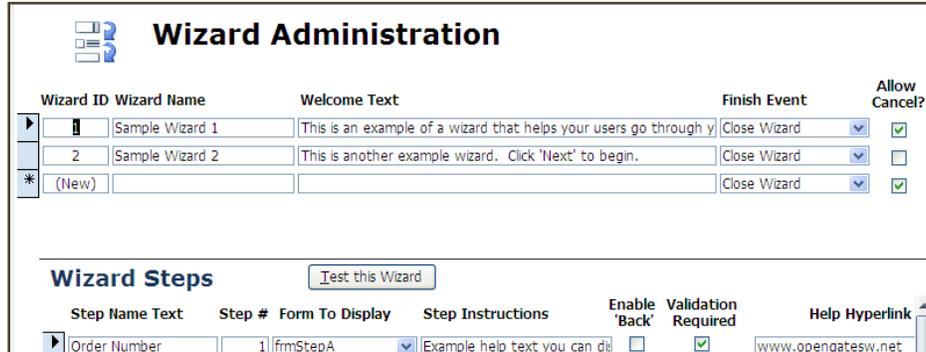


Figure 13: Wizard Administration Form

- Wizard ID** An automatically assigned identifier. Use this identifier when launching the wizard from VB code.
- Wizard Name** Identifies the common name for the wizard, which will also be displayed in the Title Bar of the wizard form.
- Welcome Text** The text that is initially displayed to the user when the wizard is launched.



- Finish Event** Determines what action Workflow Builder should take when the user clicks the “Finish” button.
- Allow Cancel** Determines whether to enable the Cancel button on the wizard form.

5.1.2 CONFIGURING WIZARD STEPS

After setting the general wizard options, you must define specific steps for the wizard to follow when the user clicks the Next button.

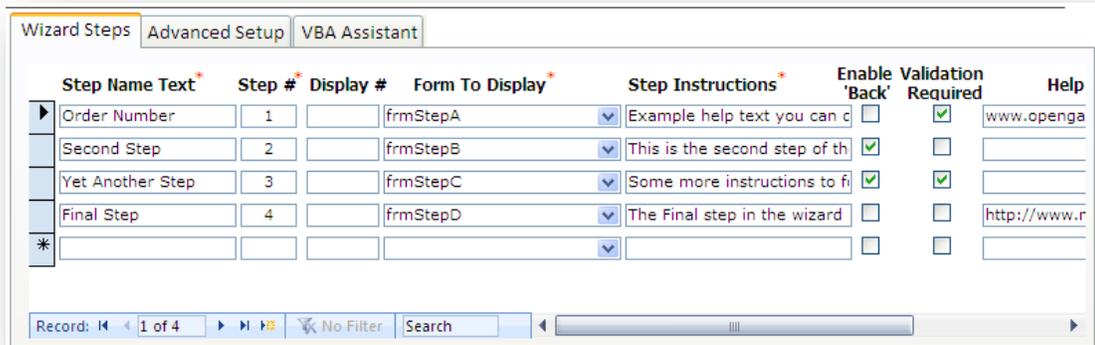


Figure 14: Wizard Step Setup

Note

You can launch the Wizard Step Editor, which provides greater screen space and visibility, by clicking on the Advanced Setup tab.

Wizard Steps Tab

- Step Name Text** The name of the step that will be displayed to the user.
- Step Number** The number of the step that you assign, and which determines when the step will be presented to the user.
- Display #** An optional step number to display on the wizard form. For example, 1A, 2D, etc. Recommended that you use the “Maximum Steps” in the Advanced Setup tab when using the Display #.
- Form to Display** The form that should be displayed for the step.
- Step Instructions** Any detailed instructions you wish to display to the user for the step in the wizard form.
- Enable ‘Back’** Determines whether the user will be allowed to return to a previous step.



Validation Required If checked, Workflow Builder will require a global variable **gvarWizardProceed** to be set to True (-1) by your form for the user to proceed. By doing so, you can ensure your users must complete all required fields on your form, if needed, before they can proceed.

Help Hyperlink The URL of a site that users can visit for additional information related to the current step. If the Help Hyperlink field is populated, the user will be able to click on a text line "Click here for additional assistance" and a browser will open using the URL you provide.

Advanced Setup Tab

Maximum Steps Optional. If you have a wizard that may branch to other wizards, specifying the maximum number of steps a user is anticipated to go through will help avoid user frustration.

5.2 REQUIRE VALIDATION (OPTIONAL)

If you wish to require a user to complete one or more fields on your form that is displayed by the Workflow Builder wizard, you must first check the "Validation Required" box for the step in the Wizard Step definition. Next, you must determine what user activities must occur in your form to allow the user to proceed to the next step of the wizard. Finally, you must add the necessary validation code within your form to check that the required conditions are met, and if so, set the global variable **gvarWizardProceed** equal to True (-1) when the user meets those conditions.

We suggest you refer to the example wizard form "frmStepB" in the Workflow Builder database as delivered by OpenGate Software. In this form, you will note that there is an OnChange event that checks to see if the dropdown box for "Select an Order" is populated. If so, **gvarWizardProceed** is set to True (-1).

5.3 SKIPPING STEPS (OPTIONAL)

If desired, you may want to skip steps within the defined wizard flow when a specific condition is met. For example, if the user checks a box "Same as Mailing Address" in an address entry form, you may want to let the user skip past the next defined form "Shipping Address." To do so, your code must set the variable **gIngSkipStep** equal to the Step Number of the step you want the user to be sent to when they click the Next button. For example, if the user is on step 3, you may want to set **gIngSkipStep** = 9 to let them skip forward to step 9 of your defined wizard process.



5.4 BRANCHING TO ANOTHER WIZARD (OPTIONAL)

If desired, you may want to branch to another wizard when a specific condition is met. For example, if the user checks a box "New Customer" in an entry form, you may want to let the user skip past the next defined form "Shipping Address." To do so, your code must set the variable **glngSkipWizard** equal to the WizardID of the wizard you want the user to be sent to when they click the Next button. For example, if the user is on wizard 3, you may want to set **glngSkipWizard** = 19 to move them to wizard 19. By default, the user will be taken to the first step of the branch wizard. You can send them to a specific step by setting the **glngSkipStep** variable as outlined in 4.3 above. To return to a specific wizard after the user has completed the branch wizard, you can set **glngReturnWizard**. To bring them to a specific step in the return wizard, set the variable **glngReturnWizardStep** the step in the return wizard you wish them to see next.

Important!

Workflow Builder maintains a history of the user's flow through one or more wizards. However, once they select the "Back" button, their forward path is once again determined by your application. For example, at Workflow 8 Step 1 their choice in an option group determines if they will proceed through Workflow 8 Step 2, or move to Wizard 21 Step 4. If they click the "Back" button and return to Worklow 8 Step 1, their option group choice will once again determine which path they take.

Additionally, if you have set **glngReturnWizard** and the user returns to the specified wizard, **glngReturnWizard** will be set to 0. The user will need to return the branch point where **glngReturnWizard** was set to proceed as expected.

5.5 LAUNCHING A WIZARD

To launch your wizards from code, use the following:

fRunWizard **WizardID**, "**FormName**", "**SubFormObject**",

IngWizardID – The WizardID for the wizard you want to run. This can be found in the first field in the Wizard Administration form.

FormName – Optional. If you plan to display your wizard forms as subforms within the frmMain window, or another form, you must provide the name of the parent form. Otherwise, Workflow Builder will display each of your forms in their own form windows.

SubFormObject – Required only if FormName is supplied. This value should be set to the name of the subform object within your parent form. For example, in frmMain, the subform object is "SubForm1"



strAnchorPoint – Optional. This value can be set to “Top Left” or “Top Right” to anchor the wizard on startup to a specific area of the screen. If not specified, the wizard will start at the top left of the screen.

5.6 EXAMPLE WIZARDS

The example wizards provide examples of the advanced methods described in the sections above. You may safely delete all the example wizard forms if desired.

6 Troubleshooting Workflows

6.1 WORKFLOW LOGS

Each workflow process is extensively logged by Workflow Builder. The logs locations and levels can be configured using the Log Administration form available via the UI Builder Administration form, or by launching form frmLogAdmin directly. Log data is, by default, stored in the table tblEvent log.

Figure 15: Log Administration screen

Note that the Logging Level setting determines whether the event log data is highly detailed (“Debug”) or contains only key workflow processing information (“Normal”).

By setting the Logging Level to debug, you can review the processing actions for a given workflow in great detail.

6.1.1 SEQUENCE IDENTIFIERS

Each workflow is given a sequence identifier at the time the workflow begins. The sequence identifier is associated with each event log entry that is created during the workflow processing. You can use the



sequence identifier to filter event log records to display only those associated with a given run of your workflow.

Important!

Some ancillary functions called during the workflow processing may not record the sequence identifier. These are supporting functions that are not necessarily specific to workflow processing, such as the fTableRecordCount function.

6.2 WORKFLOW SETUP ISSUES

The validation functionality within Workflow Builder is intended to capture most issues prior to running your workflow. However, in some cases, the validation functionality cannot identify potential errors. For example, if Step 3 of your workflow is to import data into a new table, Step 5 is to rename the table, and Step 8 is to delete the table, Workflow Builder does not attempt to ensure you keep the table names consistent between steps.

OpenGate Software recommends you review the event logs or Workflow Console carefully should a workflow process fail on a consistent basis.

7 Release History

Workflow Builder 1.0

- Initial release

Workflow Builder 1.1

- Added step resorting
- Introduced string replacement functionality
- Sequence ID logging

Workflow Builder 1.2

- Ability to import an entire directory using wildcard * character

Workflow Builder 1.3

- Added full failure actions
- Modified export actions to include queries

Workflow Builder 1.4

- System tables are now hidden from the copy/delete/rename table actions
- Fixed issue with string replacement where dates were represented by single digits instead of padded double digits (e.g., March 3rd)



should be "03" instead of "3" when replacing the [DAY] string in a workflow action parameter).

Workflow Builder 2.0 (GA 04.28.2008)

- Upgraded UI Builder components to UI Builder Enterprise Edition 3.2
- Enhancements
 - Ability to automatically run workflows on startup
 - New table-driven Wizard
 - Ability to pass values from a form or code to a Workflow to use
 - New Workflow Action options:
 - Shut down Access
 - Email Reports in PDF format (Access 2007 required)
 - Create Microsoft Outlook tasks
 - Delete records from a table based on defined criteria
 - Update records from a table based on defined criteria
 - Detect record changes in a table and notify individuals via email
 - New sample workflows triggered when records are added to a form (frmSilentWorkflowExample)
 - New string replacement [DBPath] to insert the current database's directory into a workflow command.

Workflow Builder 2.1 (GA 05.05.2008)

- Enhancements
 - Usability enhancements to the Workflow and Wizard Administration forms.
 - New VBA Assistants for the Workflow and Wizard Administration forms.
 - New "Notes" tabs for Workflows and Wizards.
 - More screen space allocated to steps.
 - Wizard enhancements
 - Ability to anchor wizards on startup.
 - Ability to branch to alternate wizards.
 - Advanced "Back" history.
 - Optional complex step numbering/naming that is distinct from the step order.
 - New larger Wizard Step Editor can be launched from "Advanced Setup" tab of Wizard Administration screen, or directly via "frmWizardStepAdminDetail"
 - New Workflow Action options
 - Open File
 - Clear Table
 - New string replacements [Param1], [Param2],[Param3], [FileOpen],[BrowseFolder]



- Adding more debug logging detail to the AutoRun functionality.
- Ability to import/export Workflow definitions and steps.
- Defect Resolutions
 - Validation on Update Table action incorrectly required a SQL Where statement when it is not required.
 - Resolved issue where multiple string replacements only reflected the last string replacement keyword provided.
 - Resolved issue where the AutoRun form would fail to execute when the VB project lost its state.
 - Validation would occur even if the Workflow had no steps with Step-Order assigned.