tools4ever

User Management Resource Administrator

UMRA Example Projects

Printer Queue Management

DISCLAIMER - Tools4ever will not be held responsible for the outcome or consequences resulting from your actions or usage of the informational material contained in this user guide. Responsibility for the use of any and all information contained in this user guide is strictly and solely the responsibility of that of the user.

Table of contents

Table of contents i

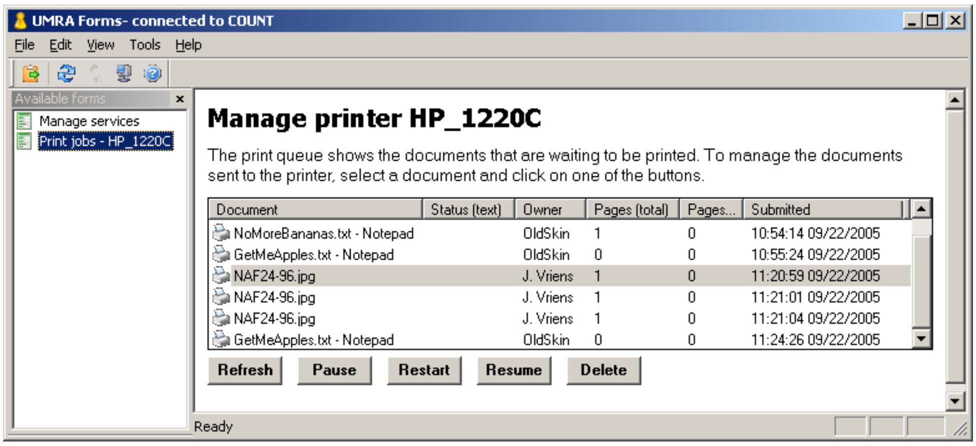
Introduction

With User Management Resource Administrator (UMRA) you can let the helpdesk manage printer queues and print jobs. Individual print jobs can be paused, restarted, resumed and deleted. The printer spooler service itself can be restarted.

This example project shows you how to setup UMRA form projects to implement this type of functionality.

In the example project, a form shows the printer documents of a particular printer. From the list, a document can be selected. The user can then press a button to pause, restart, resume or delete the print job.

UMRA Forms result



Manage printer HP_1220C

The print queue shows the documents that are waiting to be printed. To manage the documents sent to the printer, select a document and click on one of the buttons.

Document	Status (text)	Owner	Pages (total)	Pages...	Submitted
NoMoreBananas.txt - Notepad		OldSkin	1	0	10:54:14 09/22/2005
GetMeApples.txt - Notepad		OldSkin	0	0	10:55:24 09/22/2005
NAF24-96.jpg		J. Vriens	1	0	11:20:59 09/22/2005
NAF24-96.jpg		J. Vriens	1	0	11:21:01 09/22/2005
NAF24-96.jpg		J. Vriens	1	0	11:21:04 09/22/2005
GetMeApples.txt - Notepad		OldSkin	0	0	11:24:26 09/22/2005

Figure: UMRA Forms application running the example project

This document describes the main implementation aspects of this example project. The example project is also available from the Tools4ever web-site.

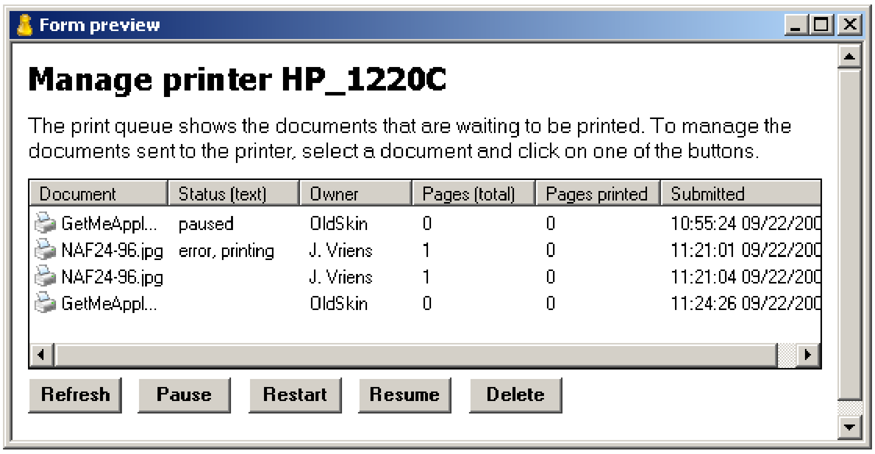
The reader of this document is supposed to be familiar with the UMRA Console, Forms and Service applications and the basics of UMRA Form projects.

UMRA projects

The example uses 2 UMRA forms project:

Main project

Print jobs – HP_1220C: The main project that contains the form and the script that is executed when one of the buttons is pressed.



Manage printer HP_1220C

The print queue shows the documents that are waiting to be printed. To manage the documents sent to the printer, select a document and click on one of the buttons.

Document	Status (text)	Owner	Pages (total)	Pages printed	Submitted
GetMeAppl...	paused	OldSkin	0	0	10:55:24 09/22/200
NAF24-96.jpg	error, printing	J. Vriens	1	0	11:21:01 09/22/200
NAF24-96.jpg		J. Vriens	1	0	11:21:04 09/22/200
GetMeAppl...		OldSkin	0	0	11:24:26 09/22/200

Figure: Form of project **Print jobs – HP_1220C**

The form contains some introduction static text fields, the main table showing print documents and the submit buttons.

Helper project

Print job list – HP_1220C: A simple project used to collect the printer documents and store the information in a table variable.

Projects principle

The main project **Print jobs – HP_1220C** shows the table with printer documents. The table data is obtained from a variable that is generated in the other project **Print job list – HP_1220C**. This project is set as the initial project of the main project.

When the user selects a document from the table, the **DocumentID** of the document is stored in a variable. Note that this **DocumentID** field is part of the table but shown in a column with a width of 0%.

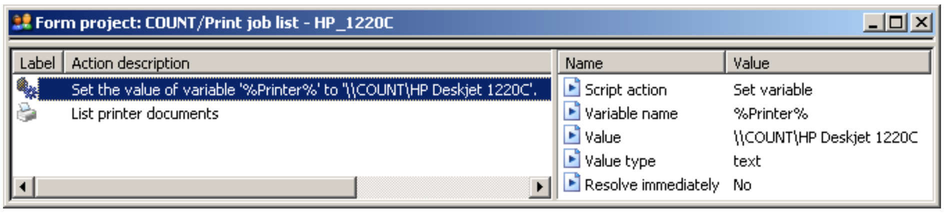
Next, the user presses one of the buttons. A variable is set, referring to the button pressed and this information, together with the **DocumentID** is sent to the UMRA Service.

The UMRA Service processes and executes the command, collects the new printer document information and returns the same form.

Helper project – Print job list – HP_1220C

The helper project only contains a script, no form. The script is executed as the initial project of main project **Print jobs – HP_1220C**. The script only contains 2 lines.

The first script action sets the value of variable **%Printer%**. The value must equal the name of a network printer (queue). Syntax of this value is: \\name_of_computer\nam_of_printer . Example: \\COUNT\HP DeskJet 1220C.



Form project: COUNT/Print job list - HP_1220C

Label	Action description
	Set the value of variable '%Printer%' to '\\COUNT\HP Deskjet 1220C'.
	List printer documents

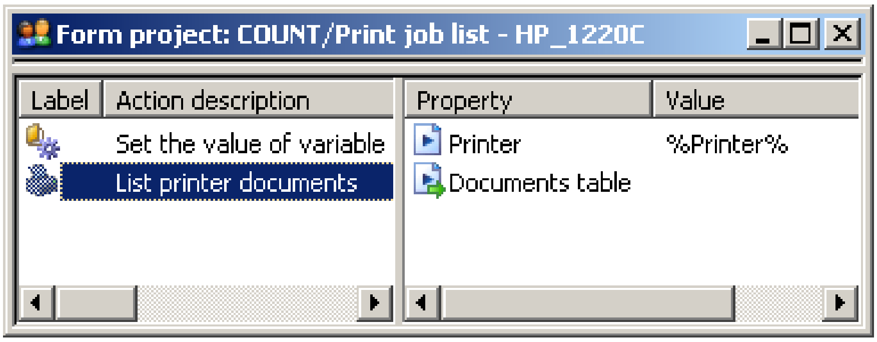
Name	Value
Script action	Set variable
Variable name	%Printer%
Value	\\COUNT\HP Deskjet 1220C
Value type	text
Resolve immediately	No

Figure: Set variable %Printer% to printer of interest

Environment specific

In another network environment, this variable must be set to the name of the printer of interest. The variable is used in the other script action and in the other main form project.

The 2nd action collects the list with printer documents.



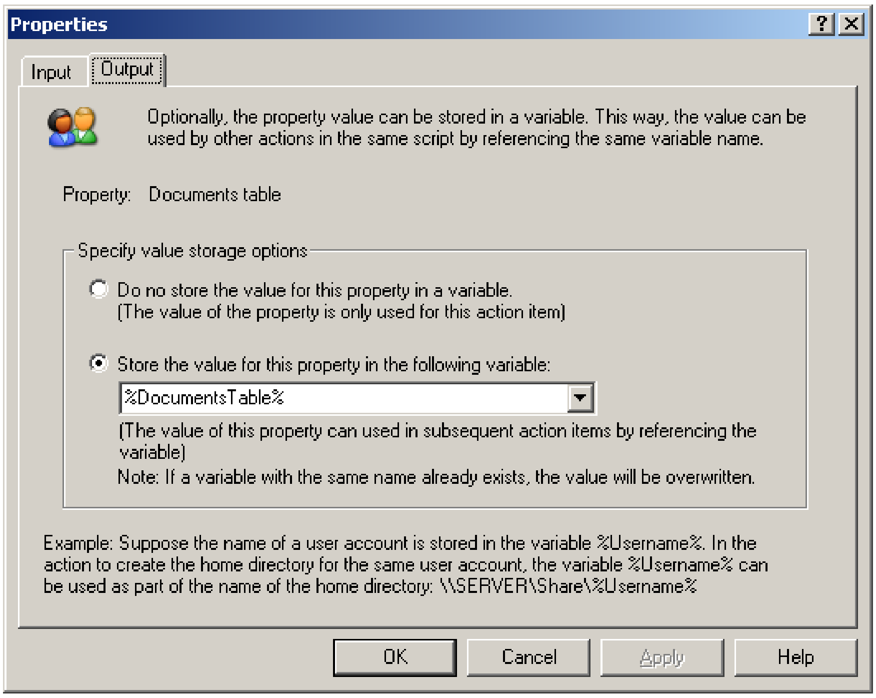
Form project: COUNT/Print job list - HP_1220C

Label	Action description
	Set the value of variable
	List printer documents

Property	Value
Printer	%Printer%
Documents table	

Figure: Action - Get printer documents info and store in table variable

The action collects the list with printer documents from the specified **%Printer%**. The result is a table that is stored in a variable specified for property **Documents table**.



Optionally, the property value can be stored in a variable. This way, the value can be used by other actions in the same script by referencing the same variable name.

Property: Documents table

Specify value storage options

○ Do no store the value for this property in a variable.
(The value of the property is only used for this action item)

● Store the value for this property in the following variable:

%DocumentsTable%

(The value of this property can used in subsequent action items by referencing the variable)
Note: If a variable with the same name already exists, the value will be overwritten.

Example: Suppose the name of a user account is stored in the variable %Username%. In the action to create the home directory for the same user account, the variable %Username% can be used as part of the name of the home directory: \\SERVER\Share\%Username%

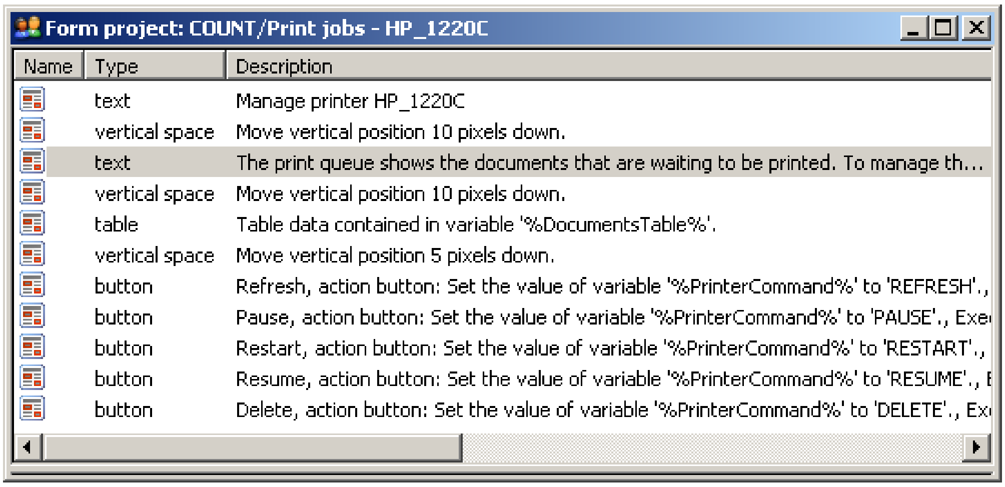
Figure: Store table with documents info in variable **%DocumentsTable%**

Output only property

The property is an output only property: the property is not needed as an input value to execute the action. Instead, the property is used to hold the output values of the action.

Print jobs project - Form

The form of the main project **Print jobs – HP_1220C** contains some introduction static text fields, the main table showing print documents and the submit buttons.



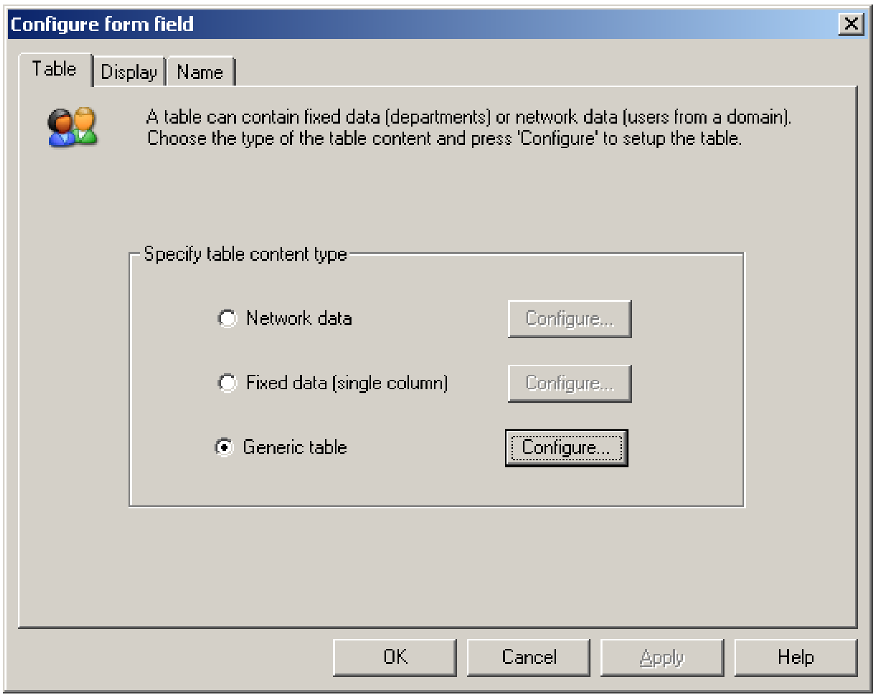
Name	Type	Description
▤	text	Manage printer HP_1220C
▤	vertical space	Move vertical position 10 pixels down.
▤	text	The print queue shows the documents that are waiting to be printed. To manage th...
▤	vertical space	Move vertical position 10 pixels down.
▤	table	Table data contained in variable '%DocumentsTable%'.
▤	vertical space	Move vertical position 5 pixels down.
▤	button	Refresh, action button: Set the value of variable '%PrinterCommand%' to 'REFRESH'.,
▤	button	Pause, action button: Set the value of variable '%PrinterCommand%' to 'PAUSE'., Exe
▤	button	Restart, action button: Set the value of variable '%PrinterCommand%' to 'RESTART'.,
▤	button	Resume, action button: Set the value of variable '%PrinterCommand%' to 'RESUME'., E
▤	button	Delete, action button: Set the value of variable '%PrinterCommand%' to 'DELETE'., Ex

Figure: Design of the main form

Except for the table and buttons, the fields are easy to implement and customize. The table and buttons fields are explained in the next topics.

Print jobs project – Table with printer documents

The table with printer documents is configured as a **generic table**, specified by a variable.



Configure form field

A table can contain fixed data (departments) or network data (users from a domain). Choose the type of the table content and press 'Configure' to setup the table.

Specify table content type

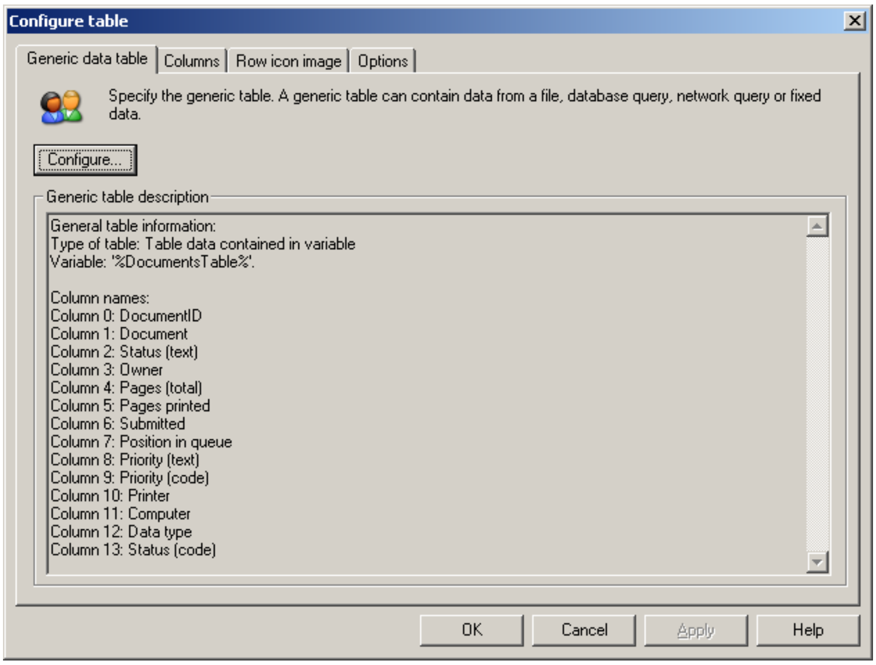
○ Network data

○ Fixed data (single column)

● Generic table

Figure: Table form field configured as **Generic table**

Click **Configure...** . In the **Configure table** window, click **Configure**.



Configure table

Specify the generic table. A generic table can contain data from a file, database query, network query or fixed data.

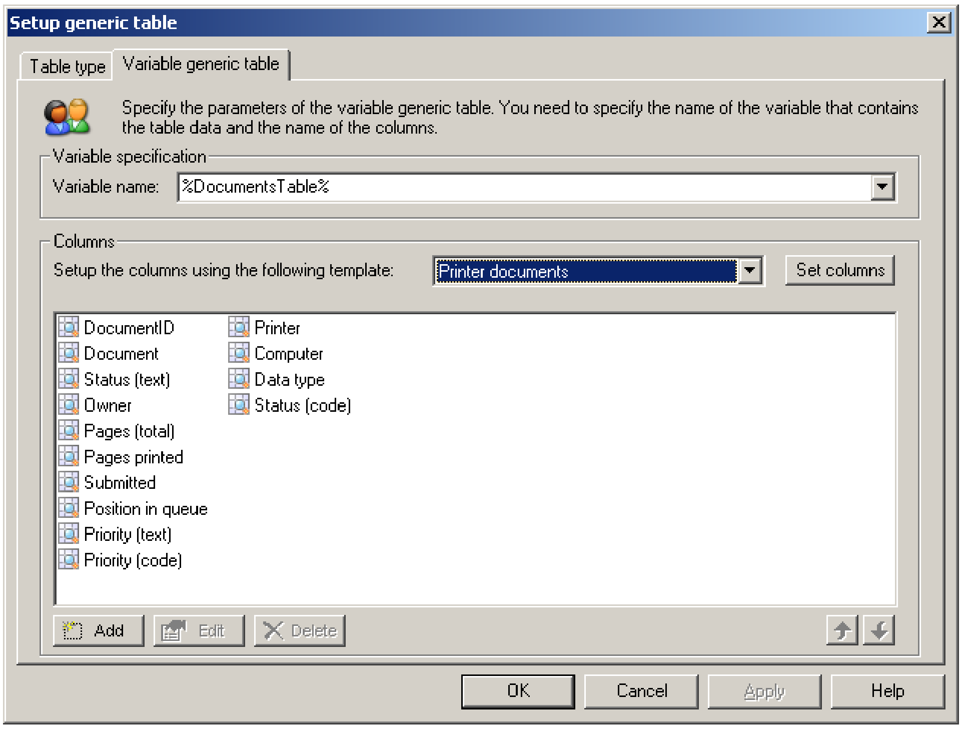
Generic table description

General table information:
Type of table: Table data contained in variable
Variable: '%DocumentsTable%'.

Column names:
Column 0: DocumentID
Column 1: Document
Column 2: Status (text)
Column 3: Owner
Column 4: Pages (total)
Column 5: Pages printed
Column 6: Submitted
Column 7: Position in queue
Column 8: Priority (text)
Column 9: Priority (code)
Column 10: Printer
Column 11: Computer
Column 12: Data type
Column 13: Status (code)

Figure: Configure table window

Select the table type as **Variable** and click on tab **Variable generic table**.



Setup generic table

Table type | Variable generic table

Specify the parameters of the variable generic table. You need to specify the name of the variable that contains the table data and the name of the columns.

Variable specification

Variable name: %DocumentsTable%

Columns

Setup the columns using the following template: Printer documents

- DocumentID
- Document
- Status (text)
- Owner
- Pages (total)
- Pages printed
- Submitted
- Position in queue
- Priority (text)
- Priority (code)
- Printer
- Computer
- Data type
- Status (code)

Figure: Setup variable generic table

Table variable specification

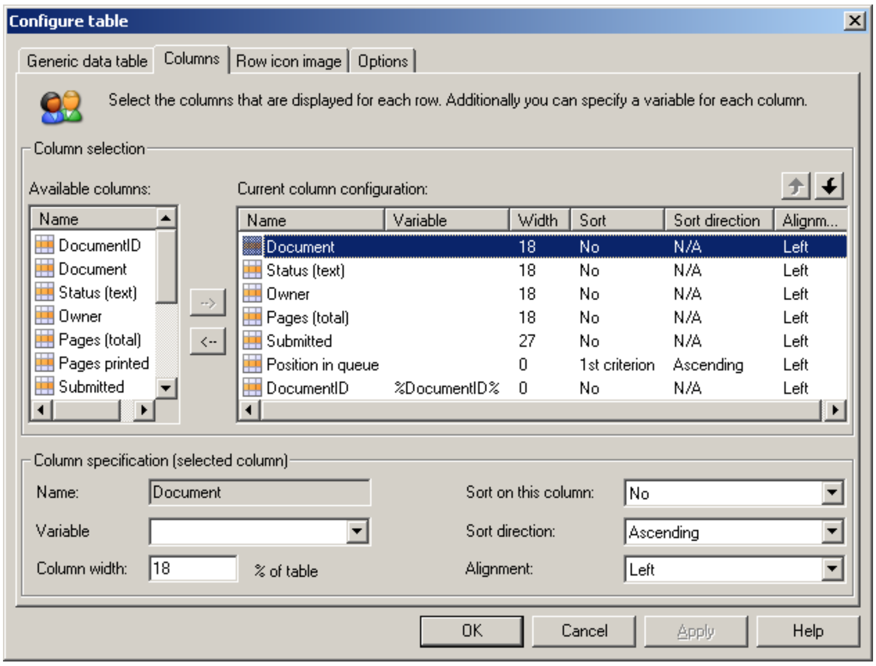
The window is used to specify the name of the variable that holds the table data and to setup the columns of the table. The **variable** name must equal the name of the variable generated by the helper project: **%DocumentsTable%**. Note that this project has no knowledge of the columns that exist for the table. That's why you need to specify the names of the columns.

Table columns specification

The table data is generated by the action **List printer documents** of helper project **Print job list – HP_1220C**. Since we know that, we know what columns are contained by the resulting table data variable. Since the action always generates the same columns, you can select the template **Printer documents** and press **Set columns** to setup the column configuration.

Note that this column information is only used to give a meaning to the columns of table data variable **%DocumentsTable%**. The columns do not necessarily correspond with the columns shown in the form. Press **OK** to continue.

To setup the columns that are shown in the form, select tab **Columns** in the window **Configure table**.



Select the columns that are displayed for each row. Additionally you can specify a variable for each column.

Column selection

Available columns:

Name
DocumentID
Document
Status (text)
Owner
Pages (total)
Pages printed
Submitted

Current column configuration:

Name	Variable	Width	Sort	Sort direction	Alignm...
Document		18	No	N/A	Left
Status (text)		18	No	N/A	Left
Owner		18	No	N/A	Left
Pages (total)		18	No	N/A	Left
Submitted		27	No	N/A	Left
Position in queue		0	1st criterion	Ascending	Left
DocumentID	%DocumentID%	0	No	N/A	Left

Column specification (selected column)

Name: Document

Variable

Column width: 18 % of table

Sort on this column: No

Sort direction: Ascending

Alignment: Left

Figure: Setup columns of form table and variable **%DocumentID%**

The list with **Available columns** shows the columns configured in the previous step. In the list **Current column configuration** you need to setup the column that must be shown in the form.

Form table return variable

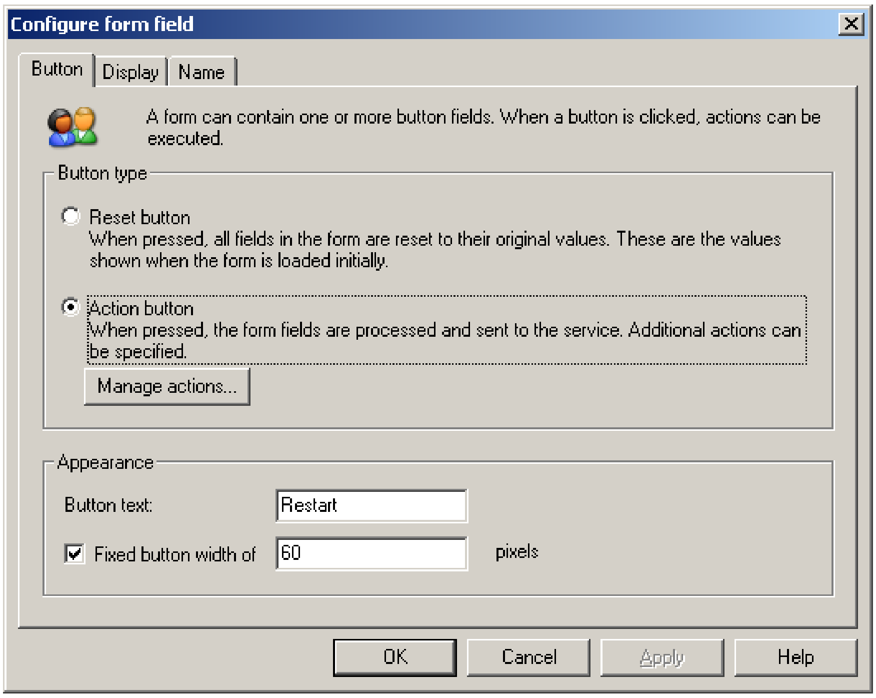
This window is also used to setup the variable that is used when the end-user selects a document from the table. This variable is used to identify the printer document that must be sent a command. The action to execute a printer command needs the id number of the document as an input property. Therefore, the variable **%DocumentID%** is configured for column **DocumentID**. Since this information is not valuable to the user, the width of the column DocumentID is set to 0%.

Now when the end-user selects a document from the list, the **DocumentID** number is stored in variable **%DocumentID%** and send to the UMRA Service for further processing.

Use the **Row icon image** and **Options** tab to fine tune the table form field with printer documents.

Print jobs project – Form buttons

The form button fields have a similar configuration. The buttons show the command to be executed:



Configure form field

A form can contain one or more button fields. When a button is clicked, actions can be executed.

Button type

○ Reset button
When pressed, all fields in the form are reset to their original values. These are the values shown when the form is loaded initially.

● Action button
When pressed, the form fields are processed and sent to the service. Additional actions can be specified.

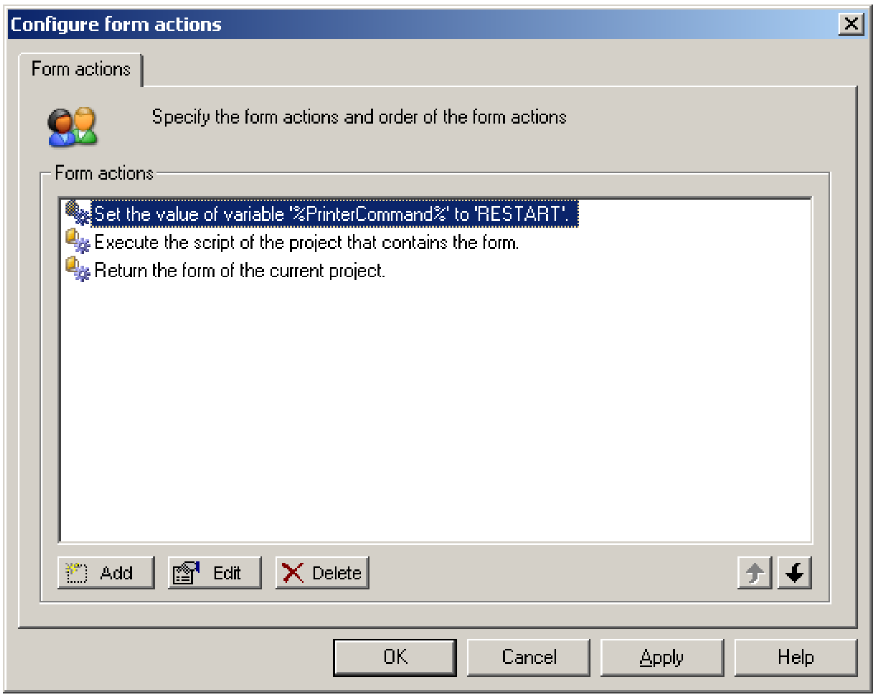
Appearance

Button text: Restart

☑ Fixed button width of 60 pixels

Figure: Button form field configuration, setup **Button text**

For each button, 3 actions are configured:



Configure form actions

Specify the form actions and order of the form actions

Form actions

- Set the value of variable '%PrinterCommand%' to 'RESTART'.
- Execute the script of the project that contains the form.
- Return the form of the current project.

Figure: Button actions

1. **Set the value of variable *%PrinterCommand%***: The variable **%PrinterCommand%** is send to the UMRA Service when the button is pressed. The variable is used in the script of the project to execute the appropriate actions.

2. **Execute the script of the project that contains the form**: The script is executed to send the printer a printer document command.

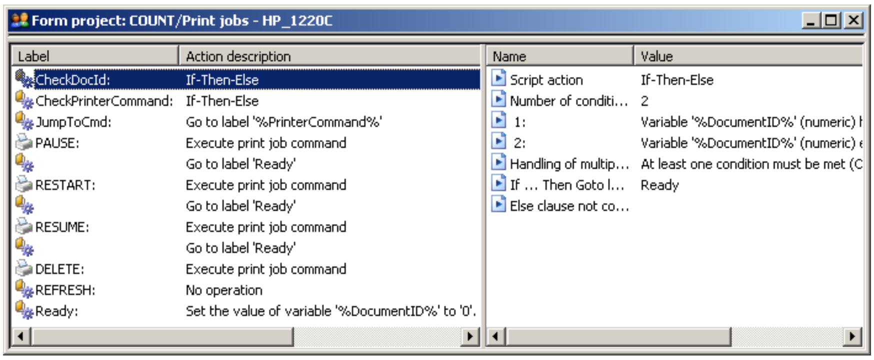
3. **Return the form of the current project**: When the printer is sent a command, the same form is returned, to reflect the status of the command sent and to allow the user to issue a command for another printer document.

Print jobs project - Script

The script of the main project **Print jobs – HP_1220C** first performs some error handling to check the end-user input. Next, control is passed to the correct action that corresponds with the button pressed.

Script action: Check %DocumentID%

In the first action, the variable **%DocumentID%** is tested to see if a document was selected from the list. If the end-user presses a submit button and no document is selected, the value either does not exist or equals zero.

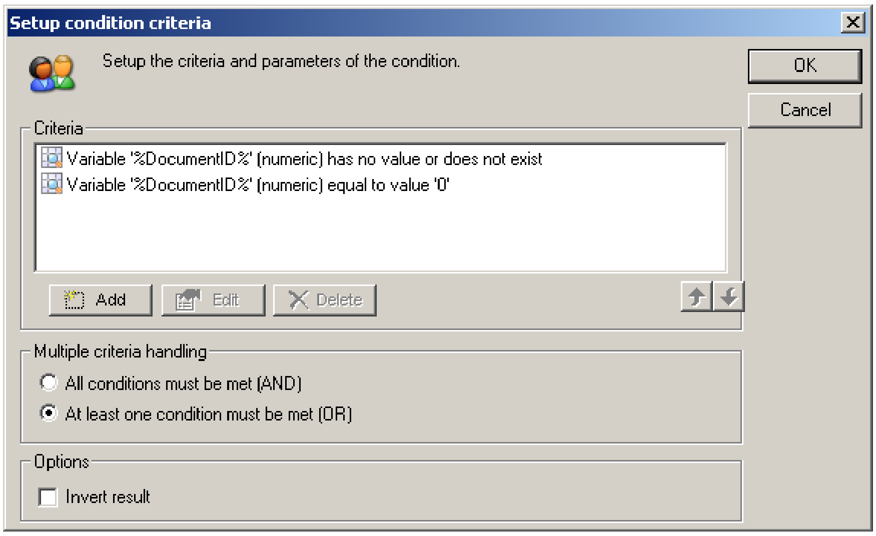


Label	Action description
CheckDocId:	If-Then-Else
CheckPrinterCommand:	If-Then-Else
JumpToCmd:	Go to label '%PrinterCommand%'
PAUSE:	Execute print job command
	Go to label 'Ready'
RESTART:	Execute print job command
	Go to label 'Ready'
RESUME:	Execute print job command
	Go to label 'Ready'
DELETE:	Execute print job command
REFRESH:	No operation
Ready:	Set the value of variable '%DocumentID%' to '0'.

Name	Value
Script action	If-Then-Else
Number of conditi…	2
1:	Variable '%DocumentID%' (numeric)
2:	Variable '%DocumentID%' (numeric)
Handling of multip…	At least one condition must be met (C
If … Then Goto l…	Ready
Else clause not co…	

Figure: Script action to check if a document was selected

The configuration of the **If-Then-Else** script action to check the value is shown below:



Setup the criteria and parameters of the condition.

Criteria

- Variable '%DocumentID%' (numeric) has no value or does not exist
- Variable '%DocumentID%' (numeric) equal to value '0'

Multiple criteria handling

○ All conditions must be met (AND)

● At least one condition must be met (OR)

Options

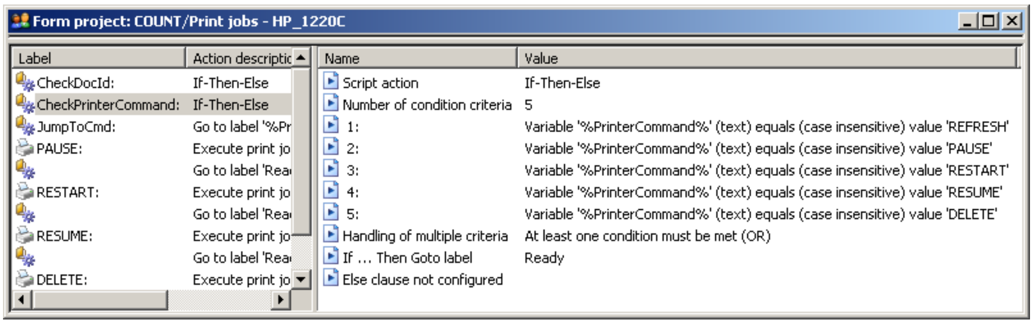
□ Invert result

Figure: Configuration of the **If-Then-Else** conditions to check the **%DocumentID%** variable

When the value is not valid, script action execution continues with the action with label **Ready**.

Script action: Check %PrinterCommand%

The next action checks the input printer command **%PrinterCommand%**. Theoretically, this action is not necessary, but it is a good habit to include thorough error-handling.

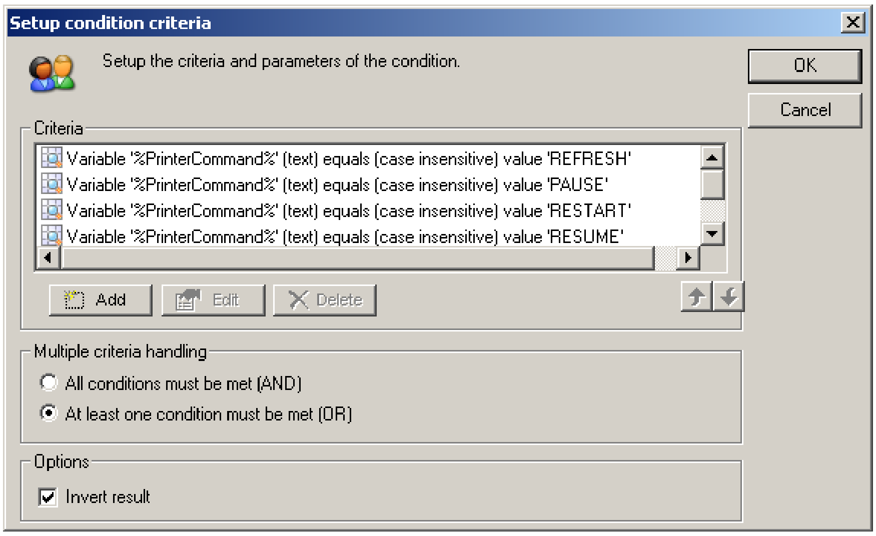


Label	Action descriptic
CheckDocId:	If-Then-Else
CheckPrinterCommand:	If-Then-Else
JumpToCmd:	Go to label '%Pr
PAUSE:	Execute print jo
	Go to label 'Rea
RESTART:	Execute print jo
	Go to label 'Rea
RESUME:	Execute print jo
	Go to label 'Rea
DELETE:	Execute print jo

Name	Value
Script action	If-Then-Else
Number of condition criteria	5
1:	Variable '%PrinterCommand%' (text) equals (case insensitive) value 'REFRESH'
2:	Variable '%PrinterCommand%' (text) equals (case insensitive) value 'PAUSE'
3:	Variable '%PrinterCommand%' (text) equals (case insensitive) value 'RESTART'
4:	Variable '%PrinterCommand%' (text) equals (case insensitive) value 'RESUME'
5:	Variable '%PrinterCommand%' (text) equals (case insensitive) value 'DELETE'
Handling of multiple criteria	At least one condition must be met (OR)
If … Then Goto label	Ready
Else clause not configured	

Figure: Script action to check input variable **%PrinterCommand%**

The action checks the value of variable **%PrinterCommand%**. If the value is not specified or does not contain one of the valid options, script action execution continues with the action with label **Ready**.



Setup condition criteria

Setup the criteria and parameters of the condition.

Criteria

- Variable '%PrinterCommand%' (text) equals (case insensitive) value 'REFRESH'
- Variable '%PrinterCommand%' (text) equals (case insensitive) value 'PAUSE'
- Variable '%PrinterCommand%' (text) equals (case insensitive) value 'RESTART'
- Variable '%PrinterCommand%' (text) equals (case insensitive) value 'RESUME'

Multiple criteria handling

○ All conditions must be met (AND)

● At least one condition must be met (OR)

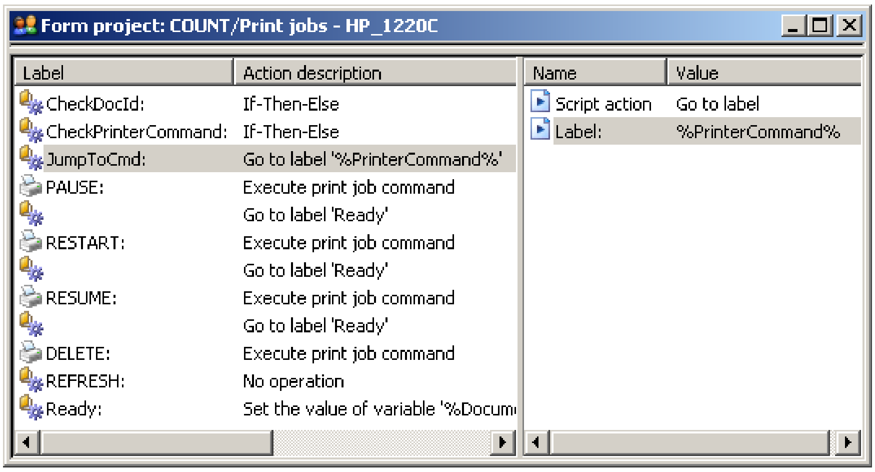
Options

☑ Invert result

Figure: Configuration of the **If-Then-Else** conditions to check the **%PrinterCommand%** variable

Script action: Go-To printer command

When the input variables are checked, the script action **Go to label %PrinterCommand%** is used to continue script action execution with the action **Execute print job command** that corresponds with the button pressed.



Form project: COUNT/Print jobs - HP_1220C

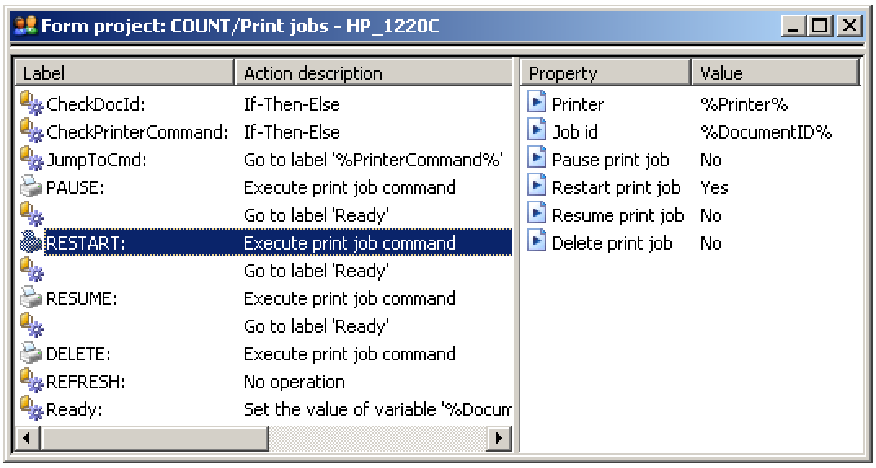
Label	Action description
CheckDocId:	If-Then-Else
CheckPrinterCommand:	If-Then-Else
JumpToCmd:	Go to label '%PrinterCommand%'
PAUSE:	Execute print job command
	Go to label 'Ready'
RESTART:	Execute print job command
	Go to label 'Ready'
RESUME:	Execute print job command
	Go to label 'Ready'
DELETE:	Execute print job command
REFRESH:	No operation
Ready:	Set the value of variable '%Docum

Name	Value
Script action	Go to label
Label:	%PrinterCommand%

Figure: Go to label %PrinterCommand% action

Now how does this work? For each of the possible printer commands, the script contains a separate section containing two script actions. The first action executes the specific printer command. The printer execution action contains a label so that the action can be jumped to. The name of the label corresponds with the value of the variable **%PrinterCommand%**. This value is set as a form action when the end user presses one of the form submit buttons. When the printer execution action is executed, the **Go to label 'Ready'** action is used to end the script.

Example: When the user presses the **Restart** button in the form, the variable **%PrinterCommand%** is set to **RESTART**. When the script is executed, the action **Go to label %PrinterCommand%** is executed as: **Go to label RESTART**. The **RESTART** label action executes the printer command to restart the printer document.



Form project: COUNT/Print jobs - HP_1220C

Label	Action description
CheckDocId:	If-Then-Else
CheckPrinterCommand:	If-Then-Else
JumpToCmd:	Go to label '%PrinterCommand%'
PAUSE:	Execute print job command
	Go to label 'Ready'
RESTART:	Execute print job command
	Go to label 'Ready'
RESUME:	Execute print job command
	Go to label 'Ready'
DELETE:	Execute print job command
REFRESH:	No operation
Ready:	Set the value of variable '%Docum

Property	Value
Printer	%Printer%
Job id	%DocumentID%
Pause print job	No
Restart print job	Yes
Resume print job	No
Delete print job	No

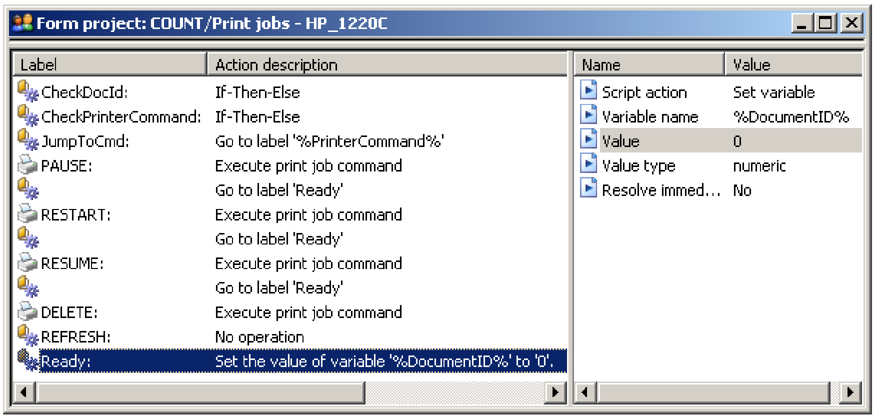
Figure: Execute print job command action

Script action: Execute print job command

The execute print job command, UMRA accesses the printer **%Printer%** as specified with helper project **Print job list – HP_1220C** and job **%DocumentID%**. This printer document ID number corresponds with the ID of the document that is selected from the table in the form.

For the **Restart** action, the property **Restart print job** is set to **Yes**. The next action jumps to label **Ready** to end the script.

When the script is completed, the next form button action is executed: **Return the form of the current project:** The cycle starts over again. As part of the form generation process, the script of the initial project **Print job list – HP_1220C** is executed.



Form project: COUNT/Print jobs - HP_1220C

Label	Action description
CheckDocId:	If-Then-Else
CheckPrinterCommand:	If-Then-Else
JumpToCmd:	Go to label '%PrinterCommand%'
PAUSE:	Execute print job command
	Go to label 'Ready'
RESTART:	Execute print job command
	Go to label 'Ready'
RESUME:	Execute print job command
	Go to label 'Ready'
DELETE:	Execute print job command
REFRESH:	No operation
Ready:	Set the value of variable '%DocumentID%' to '0'.

Name	Value
Script action	Set variable
Variable name	%DocumentID%
Value	0
Value type	numeric
Resolve immed...	No

Figure: Reset the %DocumentID% variable to 0

Script action: Reset %DocumentID%

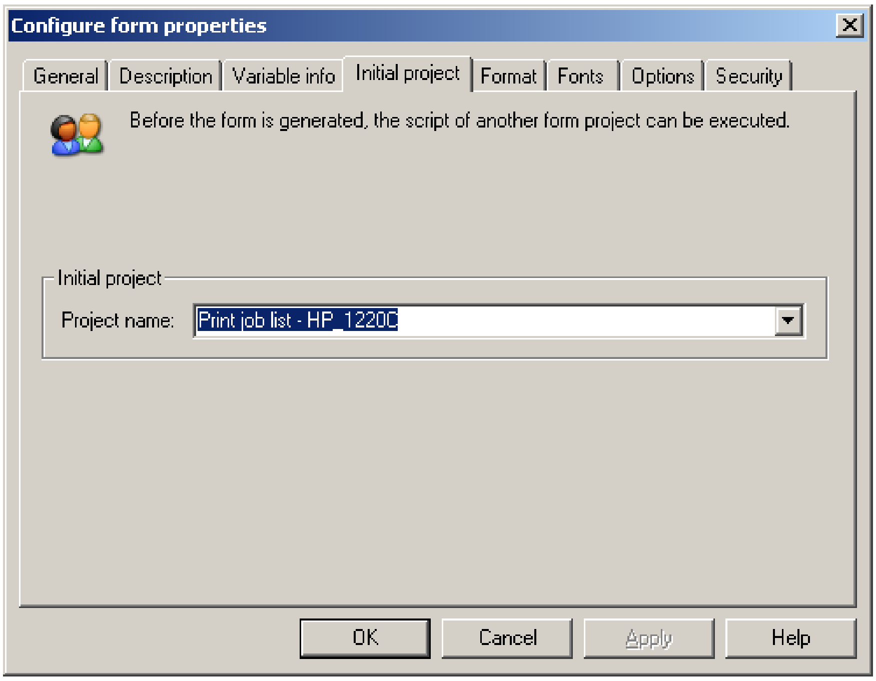
To re-initialize the %DocumentID% variable, the last action sets the value of the variable to 0. This ensures that no document is referred to when the script is executed the next time in case the end user has not selected a document.

Linking the helper and main project

The helper project **Print job list – HP_1220C** retrieves the printer documents information and stores this table information in variable **%DocumentsTable%**. The main project shows the printer documents in a table of the form of the project and sends the printer a command when a button of the form is pressed.

Initial project specification

Now how do the projects work together? The helper project is set as the initial project of the main project. When the form of the main project is generated by the UMRA Service, the script of the project **Print job list – HP_1220C** is executed first.



Configure form properties

Before the form is generated, the script of another form project can be executed.

Initial project

Project name: Print job list - HP_1220C

Figure: Initial project configuration of project **Print jobs – HP_1220C**

The helper project **Print job list – HP_1220C** collects the printer document information and stores the resulting table in variable **%DocumentTable%.** This variable is passed to the main project and used in the main table form field.

Project execution

The following section shows the log of the UMRA Service when the UMRA Forms application connects to the service, selects the form, selects a document and presses one of the submit buttons.

Project execution log

```
09:25:49 09/23/2005 Form message: '09/23/2005,09:25:49,"SSP\J.
Vriens","Forms list",OK,N/A,"2 projects found for user 'SSP\J.
Vriens'."'
09:25:53 09/23/2005 Executing form initialization project 'Print
job list - HP_1220C'.
09:25:53 09/23/2005 Variable  1: %UmraFormSubmitAccount%=SSP\J.
Vriens
09:25:53 09/23/2005 List documents in printer (queue) '\\COUNT\HP
Deskjet 1220C'.
09:25:53 09/23/2005 Form message: '09/23/2005,09:25:53,"SSP\J.
Vriens","Form load",OK,"Print jobs - HP_1220C",'
09:26:04 09/23/2005 Variable  1: %DocumentID%=6
09:26:04 09/23/2005 Variable  2: %UmraFormSubmitAccount%=SSP\J.
Vriens
09:26:04 09/23/2005 Variable  3: %Printer%=\\COUNT\HP Deskjet 1220C
09:26:04 09/23/2005 Variable  4: %DocumentsTable%=Table with 10
rows
09:26:04 09/23/2005 Variable  5: %NowDay%=23
09:26:04 09/23/2005 Variable  6: %NowMonth%=09
09:26:04 09/23/2005 Variable  7: %NowYear%=2005
09:26:04 09/23/2005 Variable  8: %NowHour%=09
09:26:04 09/23/2005 Variable  9: %NowMinute%=26
09:26:04 09/23/2005 Variable 10: %NowSecond%=04
09:26:04 09/23/2005 Variable 11: %PrinterCommand%=PAUSE
09:26:04 09/23/2005 If-Then-Else condition [Variable '%DocumentID%'
(numeric) has no value or does not exist OR Variable '%DocumentID%'
(numeric) equal to value '0'] result is FALSE, continue script
execution with next action.
09:26:04 09/23/2005 If-Then-Else condition [Variable
'%PrinterCommand%' (text) equals (case insensitive) value 'REFRESH'
OR Variable '%PrinterCommand%' (text) equals (case insensitive)
value 'PAUSE' OR Variable '%PrinterCommand%' (text) equals (case
insensitive) value 'RESTART' OR Variable '%PrinterCommand%' (text)
equals (case insensitive) value 'RESUME' OR Variable
'%PrinterCommand%' (text) equals (case insensitive) value 'DELETE'
(invert)] result is FALSE, continue script execution with next
action.
09:26:04 09/23/2005 Jump to script action with label 'PAUSE'.
09:26:04 09/23/2005 Executing command 'Pause' for job id '6' of
printer (queue) '\\COUNT\HP Deskjet 1220C'.
09:26:04 09/23/2005 Command successfully executed
09:26:04 09/23/2005 Jump to script action with label 'Ready'.
09:26:04 09/23/2005 Executing form initialization project 'Print
job list - HP_1220C'.
09:26:04 09/23/2005 Variable  1: %DocumentID%=0
09:26:04 09/23/2005 Variable  2: %UmraFormSubmitAccount%=SSP\J.
Vriens
09:26:04 09/23/2005 Variable  3: %Printer%=\\COUNT\HP Deskjet 1220C
09:26:04 09/23/2005 Variable  4: %DocumentsTable%=Table with 10
rows
09:26:04 09/23/2005 Variable  5: %NowDay%=23
09:26:04 09/23/2005 Variable  6: %NowMonth%=09
09:26:04 09/23/2005 Variable  7: %NowYear%=2005
09:26:04 09/23/2005 Variable  8: %NowHour%=09
09:26:04 09/23/2005 Variable  9: %NowMinute%=26
09:26:04 09/23/2005 Variable 10: %NowSecond%=04
09:26:04 09/23/2005 Variable 11: %PrinterCommand%=PAUSE
09:26:04 09/23/2005 List documents in printer (queue) '\\COUNT\HP
Deskjet 1220C'.
09:26:04 09/23/2005 Form message: '09/23/2005,09:26:04,"SSP\J.
Vriens","Form submit",OK,"Print jobs - HP_1220C"'
```

At 09:25:49, the forms available for the end user are loaded from the UMRA Service. At 09:25:53, the main form project is loaded. As part of the form generation process, the initial project is executed. At 09:26:04, the user selects a documents and presses the **Pause** button. The form information is submitted to the UMRA Service and the script of the project is executed: the print job is paused. Finally, the cycle starts over again: the script of the form initialization project is executed and the generated form is returned to the UMRA Forms application.

Project extensions

The example project can easily be extended to support similar functionality:

- Include documents of multiple printers in a single table
- Show multiple tables with printer documents for multiple printers
- Setup a wizard to select a printer first, and than show the printer document table of the selected printer
- Add a button to reset the printer spooler service
- Filter out documents from the table from specific users.

For additional information and other UMRA example projects, contact Tools4ever at www.tools4ever.com.

```
Filename:       UmraPrintJobManagement_v2.doc
Directory:      V:\Documentation\UMRA\Manuals\UMRA -
   Printer Queue Management
Template:       C:\Documents and
   Settings\Administrator\Application
   Data\Microsoft\Templates\Tools4ever.dot
Title:          Manage LDAP directory services with UMRA
Subject:        User's Guide
Author:         Jacques Vriens
Keywords:
Comments:       Copyright © 2005 Tools4ever bv, All righhts
   rights reserved.
Creation Date:  1/12/2006 12:03:00 PM
Change Number:  3
Last Saved On:  1/12/2006 12:05:00 PM
Last Saved By:  E. van Wezel
Total Editing Time:   2 Minutes
Last Printed On:1/12/2006 12:05:00 PM
As of Last Complete Printing
   Number of Pages:   19
   Number of Words:   2,795 (approx.)
   Number of Characters:    15,265 (approx.)
```