

**Copyrighted Material**

## **Service Manager 9 Tailoring Techniques**

**By Mike Sanders**

Copyright © 2011 by New Day Press. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system without written permission from the author, except for the inclusion of brief quotations in a review.

All product names and services identified throughout this book are trademarks or registered trademarks of their respective companies. They are used throughout this book in editorial fashion only, and for the benefit of such companies. No such uses, or the use of any trade name, is intended to convey endorsement or affiliation with the book.

ISBN 0-9708916-8-7

Printed in the United States of America

**Copyrighted Material**

## Chapter 16

# Using Wizards

The concept of scripts has been in Service Manager since its beginning, and is a basic concept in most development environments. Wizards is an advanced script technique.

Using a combination of applications and definition file records, the developer creates a program flow. This allows the user to be presented with a format accepting input and choosing an action. So, the script then chooses the proper executable code.

## What are wizards?

In Service Manager, a wizard is a collection of records, in the **wizard** file, working together to define an interaction with a user. Each **wizard** record has links allowing it to execute:

- Format control
- Processes
- JavaScript code

Wizards allow formats to be displayed either using a predefined **displayscreen** or using a custom **displayscreen**. This adds a great deal of flexibility to the user interface.

## Alternatives to wizards

Service Manager contains several tailoring tools capable of performing similar results using wizards. These are:

**Scripts.** The scripts tool is still in use out of the box when Service Manager is first installed.

Key points regarding **scripts** are:

- Provides a simple method to display a format and accept input from the user.
- Uses condition statements for determining the execution flow.
- Many modules such as change management are coded to execute scripts. These can be determined by entries in category and profile records.
- Each script execution calls a new RAD application. Which means the local variables are not passed, and loops may cause stack overflows.
- There is a predefined report printing out the entire script flow.
- Validation user input must be done with code in the script, not format control.

## Copyrighted Material

**Process.** The Process table is part of the document engine tool. It has a means for defining branching conditions just like wizards and scripts.

Key points regarding scripts are:

- Getting user input is more complex, since it must call the display application needing a great deal of variable setup.
- Process stays in a single RAD application, which means the local variables are passed to all branches.

## Uses for wizards

Wizards are excellent for getting a user input and processing it outside a normal module work flow. Usually this means buttons are created on a ticket, and the displayoption calls the **run.wizard** RAD application.

Possible uses include:

- Creating a database record.  
This involves taking data from the current ticket, and combining it with user input. Then wizards would add the record to a database. Which could be some kind of supplementary information.
- Selecting records from a database.  
This takes some data from the current ticket, and uses it for selecting records from another database. So, the user could then select a record, or initiate some action against the other database.
- Modify the current record.  
This would involve presenting the user with modification options not available on the current ticket. And could be based upon user privileges, or conditions in the ticket. With the changes put into the ticket and saved.
- Stand-alone application for creating a CI, as an out of the book wizard. In that case, the wizard is called from a menu.

## Basic scenarios

Wizards offer a very complex set of options. Which, depending on the settings, can cause the wizard to behave very differently.

Some basic scenarios are:

**Get the user's input with a single record.** This scenario displays a single database record waiting for the user to input data, and select an action. And these uses would be for updating or creating a new record.

The displayed record could be a new or existing record. And if a query selects the record, then only the first record is displayed.

## Copyrighted Material

## Copyrighted Material

**Allow user to choose one or more records from a qbe list.** This is used when the exact record, or records to be worked on, are not known.

In this case, the wizard does a query and displays the results. So the user can highlight a single record or several. And then select Next, to pass their selection on to the code in the current wizard. Or execute another wizard, where further input can be entered.

**Run Action without user input.** A wizard just executes code, if Skip Display is selected. In this case, it acts as if the user clicked Next. This would be used if the record is to be called from more than one record, otherwise the code could be in the display record.

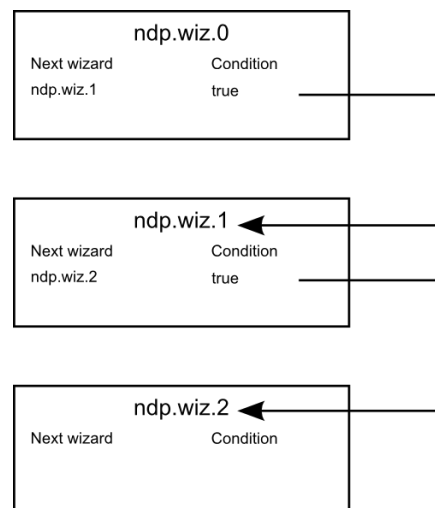
**Run Cancel without user input.** A wizard can just execute code if Cancel Immediately is selected. In this case, it acts as if the user clicked Cancel.

## Wizard work flow

A single wizard consists of one or more **wizard** records. If the **Skip Display?** checkbox is checked, then a single wizard record can be executed without any other input on the record. This would be the most simple wizard possible.

If more than one wizard record is used, then the condition section, of the initial wizard record, must be used. And it contains two arrays. One holding the names of other wizards, and the other containing condition statements.

Here is the flow for executing three wizard records sequentially.

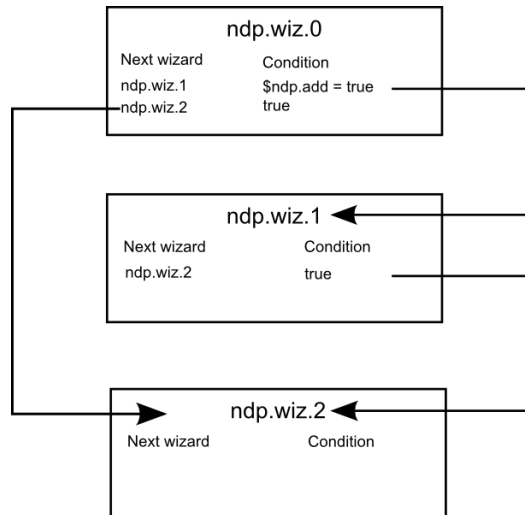


This example just uses a simple flow to move between wizard records. So, the condition is always true.

## Copyrighted Material

## Copyrighted Material

Here is a more complex example. It branches, meaning the next executed record is different, depending upon how the condition statements evaluate.



This shows after **ndpd.wiz.0** is executed and checks if the variable `$ndp.add` is set to true. If it is, then **ndp.wiz.1** is executed. If not, then **ndp.wiz.2** is executed, since it has a true condition. If the condition on **ndp.wiz.2** was something else, then there could be situations where no other wizard records are executed.

## Getting a file variable

Wizards must have a file variable called `$L.file`. Which is passed in from the **displayoption** calling **run.wizard**.

If it is being called from a ticket, then that ticket record may be passed in. But when it is passed in, it becomes **\$L.file** in the wizard.

There are four options available for a wizard.

- |                           |   |
|---------------------------|---|
| <b>No \$L.file</b>        | This option does not mean <b>\$L.file</b> does not exist. It defaults <b>\$L.file</b> to be an empty record for the typecheck file.                       |
| <b>\$L.file passed in</b> | <b>This option means a file variable is passed in using the file parameter.</b>   |
| <b>Create a record</b>    | <b>This option allows the entry of any dbdict name. With the wizard initializing \$L.file as an empty record for the file.</b>                            |
| <b>Select records</b>     | <b>This option allows the entry of any dbdict name and query, for selecting records. With the wizard initializing \$L.file, then executing the query.</b> |

## Copyrighted Material

## Displaying a format

Each wizard record can display a format to the user. And this is controlled by the checkbox labeled Skip Display?

When checked the wizard does not display a format. So, this is used when the wizard record is for processing only.

If **Skip Display?** is not checked, then a format is displayed. With wizards using a main format concept, and a subformat controlling the user interface.

It does this so the developer does not have to create formats with buttons, if the navigation is simple.

Wizards use a combination of a main format and a subformat. With the main format being predefined in the system. And the subformat creating the specific situation defined in the wizard.

### Main format


There are seven main formats defined out of the box. And they vary in layout and available buttons.

Main format	Use	Buttons
Small wizard.small	Used with low resolution interfaces	Next, Previous, Cancel, Finish
Medium wizard.medium	Used with most systems	Next, Previous, Cancel, Finish
Large wizard.large	Used when a large amount of information must be displayed	Next, Previous, Cancel, Finish
OK only – small wizard.okonly.small	Used for a simple user action with the user having a low resolution display and no subformat used	OK, Cancel
OK only – medium wizard.okonly.medium	Used for a simple user action with normal resolution and no subformat used	OK, Cancel
Buttons on top	This is the medium format, with the buttons on top	Next, Previous, Cancel, Finish
Buttons on side	This is the medium format, with the buttons on the side	Next, Previous, Cancel, Finish

### Subformat

The wizard record holds the format name format created for this wizard. And the subformat must be small enough to prevent it from overlapping the Next/Previous/Finish buttons.

---

 **NOTE** The format displayed by wizards has the file variable of \$L.file. So, this file sets as selected on the wizard record.

---

## Copyrighted Material

The subformat must not be too large, since it covers the buttons. So, best to use **Buttons On Top** for large subformats.


### Displaying qbe lists

If a selection is used for the initial file, a qbe format must also be used. With one of the **Select One** or **Select Multiple** records, from a list, selected on the **Usage** tab. So, the Initial Expressions and JavaScript are not executed, should records be selected.

#### *Flow*

- Initialize \$L.file.
- Select the records.
- Display a qbe list.
- User selects one or multiple records.

---

 **NOTE** With multiple records, double click does not work.

---

- Use control click to select out of order.
- Use shift click to select a range.
- The **Next** button executes **Actions**. With the flow going to another wizard for further processing.

If **Select one record** or **Select multiple records** from the list is used on the **Usage** tab, then the results of the defined query is displayed.

The **Subformat** entered, must be a qbe format, for the file named in the **Query record type of** field.