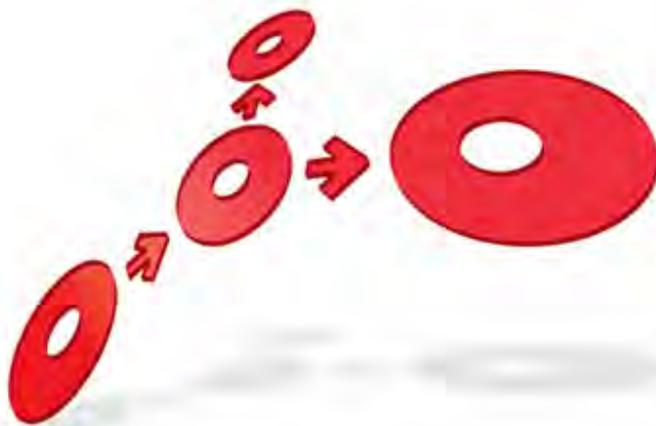


Portrait Foundation



Package Management User Guide

Edition 4.1

11 January 2013



 **Pitney Bowes**
Software



Portrait Foundation Package Management User Guide

©2013
Copyright Portrait Software International Limited

All rights reserved. This document may contain confidential and proprietary information belonging to Portrait Software plc and/or its subsidiaries and associated companies.

Portrait Software, the Portrait Software logo, Portrait, Portrait Software's Portrait brand and Million Handshakes are the trademarks of Portrait Software International Limited and may not be used or exploited in any way without the prior express written authorization of Portrait Software International Limited.

Acknowledgement of trademarks

Other product names, company names, marks, logos and symbols referenced herein may be the trademarks or registered trademarks of their registered owners.

About Portrait Software

Portrait Software is now part of [Pitney Bowes Software Inc.](http://www.pitneybowes.com)

Portrait Software enables organizations to engage with each of their customers as individuals, resulting in improved customer profitability, increased retention, reduced risk, and outstanding customer experiences. This is achieved through a suite of innovative, insight-driven applications which empower organizations to create enduring one-to-one relationships with their customers.

Portrait Software was acquired in July 2010 by Pitney Bowes to build on the broad range of capabilities at Pitney Bowes Software for helping organizations acquire, serve and grow their customer relationships more effectively. The Portrait Customer Interaction Suite combines world leading customer analytics, powerful inbound and outbound campaign management, and best-in-class business process integration to deliver real-time customer interactions that communicate precisely the right message through the right channel, at the right time.

Our 300 + customers include industry-leading organizations in customer-intensive sectors. They include 3, AAA, Bank of Tokyo Mitsubishi, Dell, Fiserv Bank Solutions, Lloyds Banking Group, Merrill Lynch, Nationwide Building Society, RACQ, RAC WA, Telenor, Tesco Bank, T-Mobile, Tryg and US Bank.

Pitney Bowes Software Inc. is a division of Pitney Bowes Inc. (NYSE: PBI).

For more information please visit: <http://www.pitneybowes.co.uk/software/>

UK

Portrait Software
The Smith Centre
The Fairmile
Henley-on-Thames
Oxfordshire, RG9 6AB, UK

Email: support@portraitsoftware.com
Tel: +44 (0)1491 416778
Fax: +44 (0)1491 416601

America

Portrait Software
125 Summer Street
16th Floor
Boston, MA 02110
USA

Email: support@portraitsoftware.com
Tel: +1 617 457 5200
Fax: +1 617 457 5299

Norway

Portrait Software
Portrait Million Handshakes AS
Maridalsveien. 87
0461 Oslo
Norway

Email: support@portraitsoftware.com
Tel: +47 22 38 91 00
Fax: +47 23 40 94 99

About this document

Purpose of document

The purpose of this document is to describe how to work with packages in the Portrait Foundation Configuration Suite.

Intended audience

The intended audience is users of Portrait Foundation Configuration Suite. A basic familiarity with Portrait Foundation configuration concepts is assumed.

Related documents

None

Software release

Portrait Foundation 3.1 or later.

Contents

1	Introduction	7
2	Concepts	8
3	Opening packages and workspaces	10
3.1	Introduction	10
3.2	Opening a package	10
3.3	Opening a workspace	10
3.4	Working with two configuration instances	11
4	Working with packages	13
4.1	Introduction	13
4.2	Package properties	13
4.3	Create a package	14
4.4	Create a workspace	14
4.5	Include a package	15
4.6	Remove a package	15
4.7	Find directly included packages	16
4.8	Find a package's configuration	16
4.9	Find a configuration object's package	17
5	Package interface	18
5.1	Publishing configuration	18
5.2	Concepts	18
5.3	User interface	19
6	Working with overlays	21
6.1	Introduction	21
6.2	Create an overlay	21
6.3	Extend an overlay	22
6.4	Delete an overlay	22
6.5	Create an override	23
6.6	Edit an override	23
6.7	Delete an override	24
7	Reference browser	26
7.1	Introduction	26
7.2	Dialog overview	26
7.3	Dialog layout	27
7.4	User interface description	28
7.5	Reference paths dialog	29
7.6	Reference browser settings dialog	31

8	Move configuration wizard	32
8.1	Introduction	32
8.2	Moving configuration	32

1 Introduction

This guide covers the following topics.

- **Package concepts** – describes packages, overlays, and related concepts. All users need to read this section.
- **Opening a package** - describes how to open a package. All users need to read this section.
- **Working with packages** – describes how to create and modify packages. Users who are responsible for creating substantial amounts of new configuration need to read this section.
- **Package interface** – describes how to set an interface on a package. Users who are responsible for publishing packages to other users need to read this section.
- **Working with overlays** – describes how to use overlays when working with supplied packages. All users need to read this section.
- **Reference browser** – describes the reference browser window. The reference browser is used in moving configuration objects from one package to another. Only expert users need to read this section.
- **Moving configuration** – describes how to move configuration objects from one package to another. Only expert users need to read this section.

2 Concepts

This document describes how configuration is organised, and how users can manage their configuration.

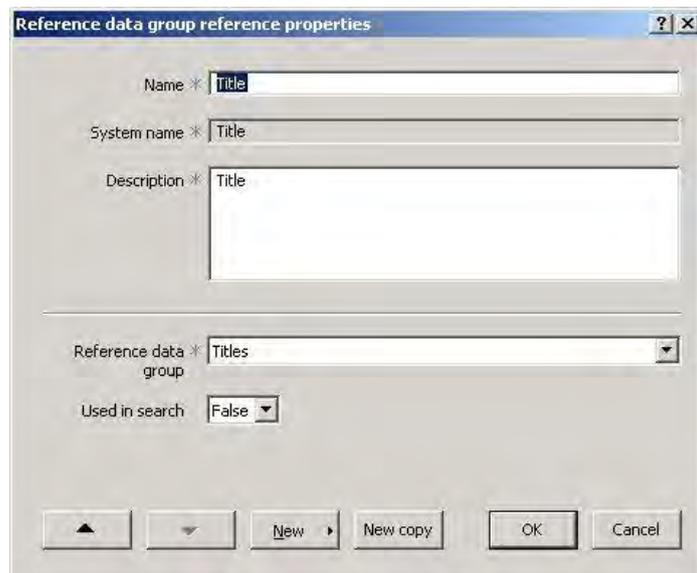
The following paragraphs describe the key concepts of **configuration objects**, **packages**, and **overlays**.

A **configuration object** is a re-usable definition. Parties, products and process models are all examples of Portrait configuration objects. In the Configuration Suite the term "object" is used to refer to configuration objects.

The definition of a configuration object is built from configuration items such as attributes, operations, inputs, outputs, and data mappings. All these items are not re-usable, and are part of the internal definition of a configuration object.

Configuration objects are linked by **references**. In the example illustrated below, the definition of the party attribute **Title** refers to the reference data group which holds a list of titles.

Figure 1 - Example of a reference



A **dependency** is a reference viewed in the other direction. In the above example, the party attribute **Title** is dependent on the reference data group **Titles**.

A **repository** is the store in which configuration objects are held. A user of Configuration Suite logs onto a repository to access the configuration he is working with.

A **package** is a unit of configuration which can be moved between repositories. Packages contain configuration objects and other packages.

A configuration object can belong to only one package.

A **workspace** is a package which can be deployed to a database. There can be more than one workspace in a repository. Only one workspace can be deployed to the same database. A workspace can contain configuration for more than one application.

A package can have a **package interface** declared on it. The purpose of a package interface is to define those parts of the configuration which are intended

to be accessed by users of the package. This means that the owner of the package can subsequently modify or remove configuration objects which are not exposed in the interface, and be sure that he will not disrupt configuration built by users of the package.

If you import a package supplied to you from another repository, you are not able to edit the content of the package in your own repository. This restriction is necessary to allow you to be able to take configuration which has been built by someone else, and later to be able to take upgrades of it. You will however still need to be able to add your own configuration to packages which you have imported. This is done using overlays.

An **overlay** holds changes to a configuration object which you have added in a different package. An overlay is also a configuration object, has references, and can be referenced. The changes held in an overlay are stored separately in the repository.

Overlays hold two types of change, extensions and overrides.

An **extension** is an extension to a configuration object which has been added in an overlay. For example, adding a new attribute or a new operation to a party object are both examples of extensions.

An **override** is a change in the value of a configuration object property which has been added in an overlay. For example, it would be possible to change the field labels in a generated interaction by overriding them. The rest of the generated interaction would work in the same way as the original.

Only some properties of a configuration object can be changed in an override.

3 Opening packages and workspaces

3.1 Introduction

This section tells you how to do the following.

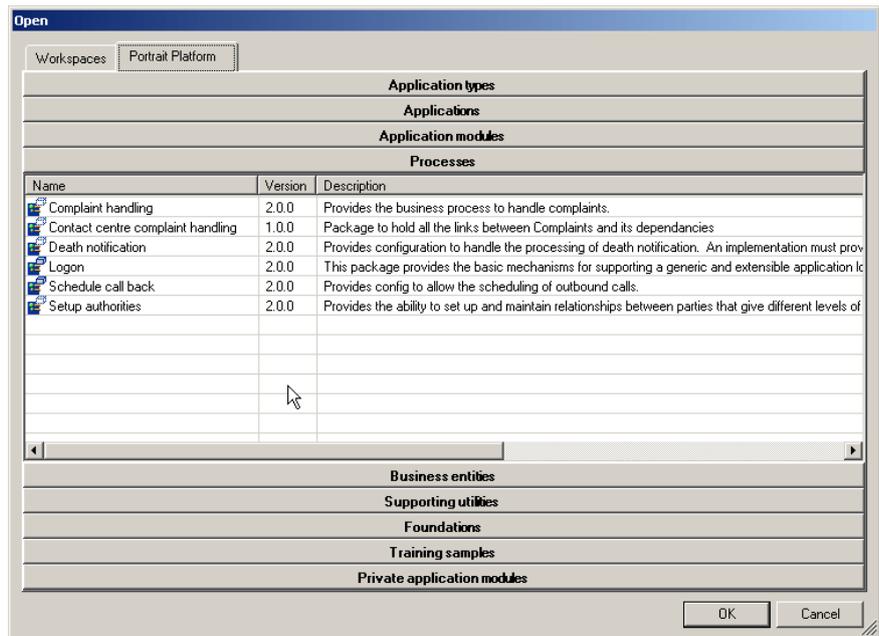
- Open a package
- Open a workspace

You can only edit a configuration object after opening the package which contains it. Similarly you can only deploy configuration after opening the workspace. You will therefore be opening packages or opening the workspace regularly while working with configuration.

3.2 Opening a package

To open a package, select **File Open** from the main menu bar. The **Open package** dialog is displayed.

Figure 2 - Open package dialog

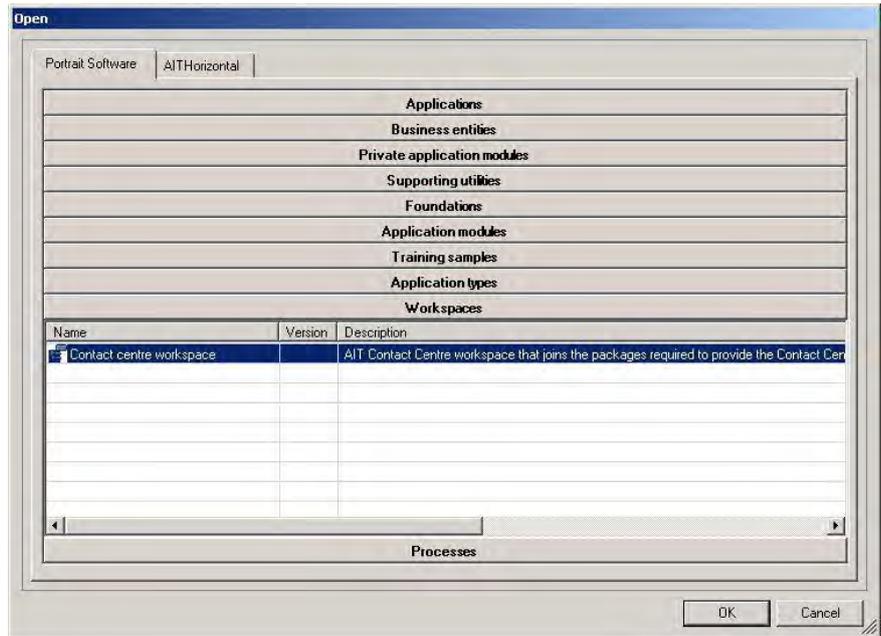


The **Open package** dialog lists all packages in the repository. There are separate tabs for each ownership in the repository. (Refer to **Repository Manager User Guide** for a description of repository ownership.) Within each tab packages are listed by classification. Open the shutter for the appropriate classification and select the required package from it.

3.3 Opening a workspace

To open a workspace, select **File Open** from the main menu bar. The package viewer dialog is displayed. Select the workspace from the **Workspace** classification.

Figure 3 - Open workspace

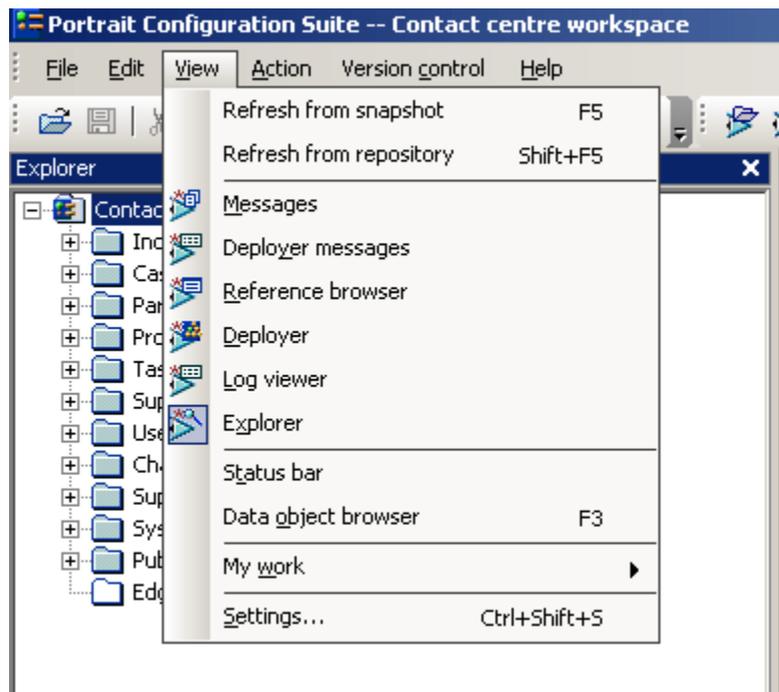


3.4 Working with two configuration instances

It is sometimes convenient to work with two configuration instances. For example, you may choose to open one instance of Configuration Suite to edit a package, and another to deploy the workspace. This is more convenient than opening the workspace each time that you make a deployment, and then re-opening the package to make further edits.

If you are doing this, use **Refresh** to refresh the other instance with the changes you have made in the first instance. In this example, after making changes in the customer package, use **Refresh** to update the workspace before deploying.

Figure 4 - Refresh menu option



4 Working with packages

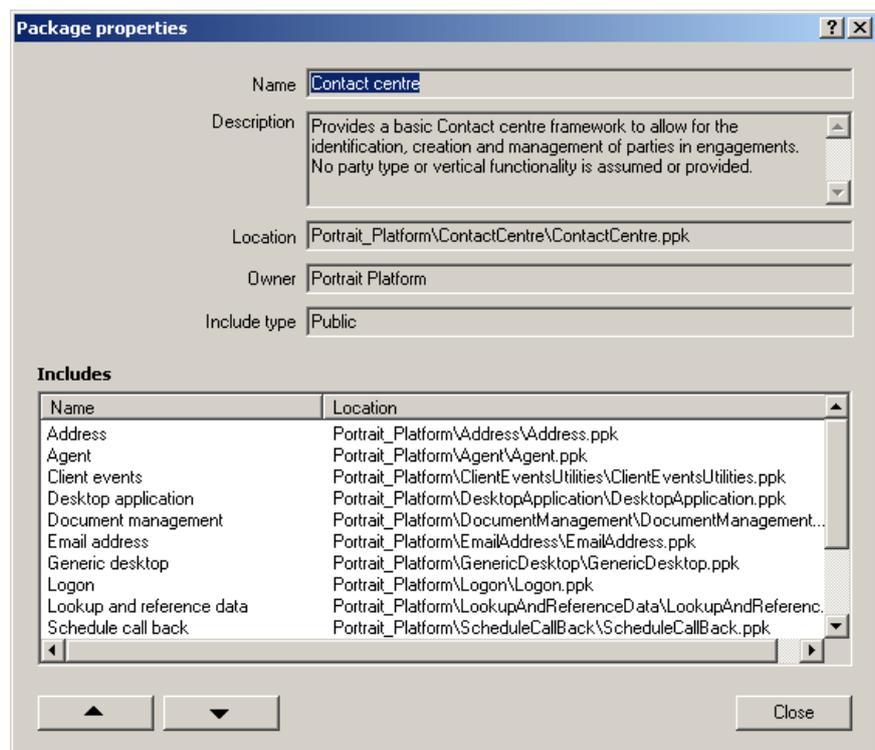
4.1 Introduction

This section tells you how to do the following.

- View the properties dialog for a package.
- Create a package
- Create a workspace
- Include a package
- Remove a package
- Find directly included packages
- Find a package's configuration
- Find a configuration object's package

4.2 Package properties

Figure 5 - Package properties dialog



The **Package properties** dialog displays the following properties.

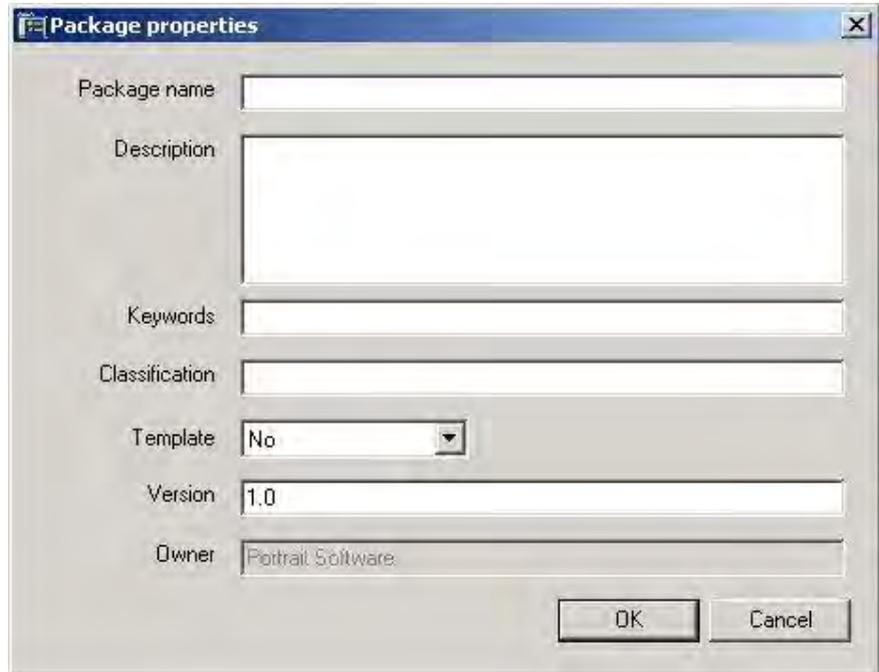
- **Name** The Name of a package cannot be changed once it has been created.
- **Description** The Description should be used to give a brief summary of the function of the package.
- **Location** The Location shows the path and filename for the package. The Location is created by the system and cannot be amended.
- **Owner** The Owner of a package is set by the ownership of the repository in which it was created. It cannot be amended.
- **Include type** The Include Type is either Public or Private. It is an attribute of the inclusion – that is, it shows whether the inclusion of the package in the currently open package is Public or Private.

- **Includes** The Includes field lists the packages which are included in this package. Only directly included packages are listed.

The Package Properties dialog is read only.

4.3 Create a package

Figure 6 - New package



The screenshot shows a dialog box titled "Package properties" with a close button (X) in the top right corner. The dialog contains several input fields and a dropdown menu:

- Package name: A text input field.
- Description: A large text area.
- Keywords: A text input field.
- Classification: A text input field.
- Template: A dropdown menu with "No" selected.
- Version: A text input field containing "1.0".
- Owner: A text input field containing "Portrait Software".

At the bottom right, there are two buttons: "OK" and "Cancel".

Select File New Package from the main menu bar. The **New package** dialog is displayed. Enter the name and other properties for the new package.

A new package is not automatically included in the current package or workspace. Use the **Add package** option in the context menu to include it.

4.4 Create a workspace

Figure 7 - New workspace



The screenshot shows a dialog box titled "Workspace properties" with a close button (X) in the top right corner. The dialog contains three input fields:

- Workspace name: A text input field.
- Description: A large text area.
- Keywords: A text input field.

At the bottom right, there are two buttons: "OK" and "Cancel".

Select File New Workspace from the main menu bar. The **New workspace** dialog is displayed. Enter a **Name** and **Description**. **Keywords** may be entered if required.

A new workspace is empty. Add configuration to the new workspace by including the required packages.

4.5 Include a package

To include a package in the current package, select the **Included packages** folder in the Explorer view.

Figure 8 - Included packages folder

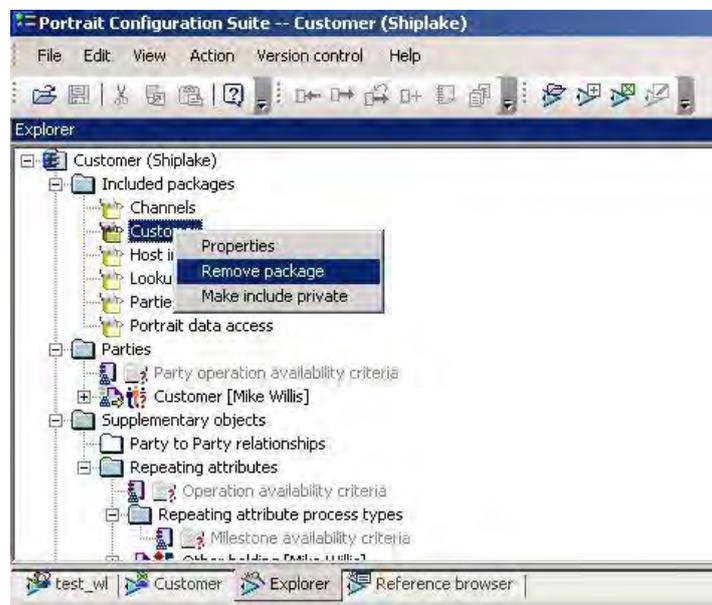


From the context menu, select the option **Add package**. The **Select package** dialog is displayed. Select the package from the appropriate classification. It will be added to the **Included packages** folder.

4.6 Remove a package

To remove a package from its inclusion in another package, select the package in the **Included packages** folder in the Explorer View. Select the **Remove package** option in the context menu.

Figure 9 - Remove package



Removing a package will not proceed if there are still references to it from the current package. Remove these references before trying to remove the package. The **Reference browser** can be used if necessary to help find the current references. (See section 7.)

The **Remove package** option is only available for packages which are directly included in the current package.

4.7 Find directly included packages

Directly included packages are those which are directly included in the currently open package. They are displayed in normal font. A directly included package can be removed, and its inclusion type can be changed.

Indirectly included packages are those which are included indirectly, through another package. They are displayed in greyed out font. An indirectly included package cannot be removed, and its inclusion type cannot be changed.

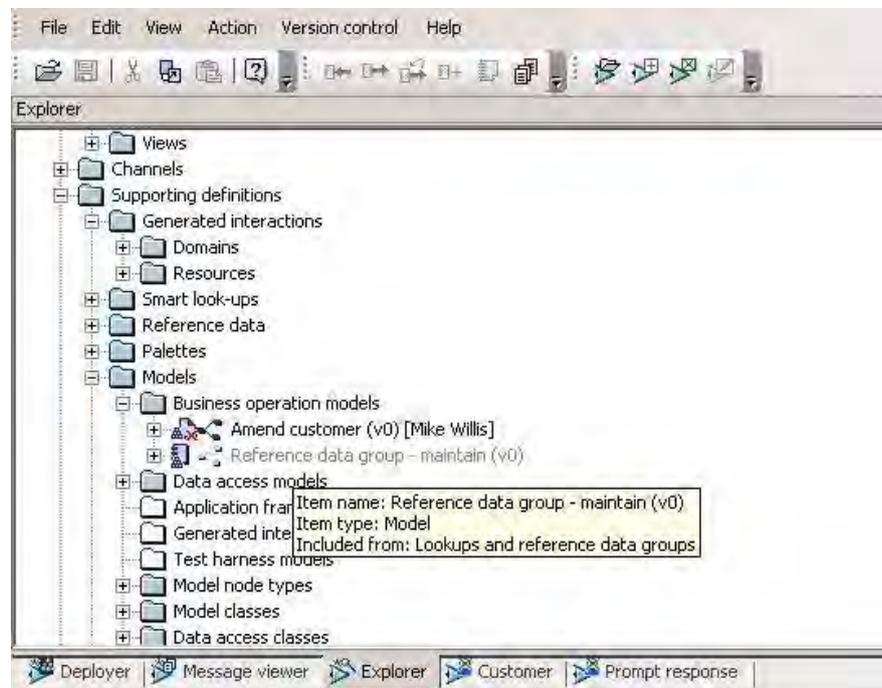
Figure 10 - Directly and indirectly included packages



4.8 Find a package's configuration

The Configuration Suite lists all configuration objects included in the current package, both its own objects and those which are included from other packages. Its own objects are distinguished by being displayed in normal font. Objects included from other packages are displayed in a greyed out font.

Figure 11 - Objects owned by a package



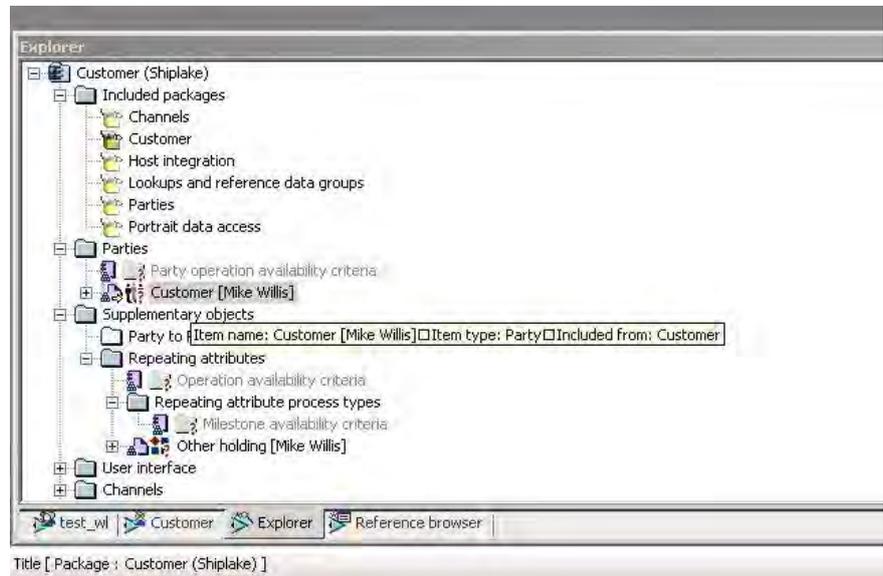
In the example above, the model **Amend customer** belongs to the current package. The model **Reference data group – maintain** has been included from another package.

4.9 Find a configuration object's package

A configuration object's package can be found in any of the following ways.

- 1 In a work folder the package name is displayed for each object.
- 2 In an Edit window or in Explorer View, select the object. The package name is displayed in the tooltip and the status bar.

Figure 12 - Configuration object tooltip



5 Package interface

5.1 Publishing configuration

It must be possible to publish configuration. This is necessary when configuration is supplied by Portrait Software or a partner to users of a Portrait Foundation system. Taking pre-built configuration in this way cuts costs, time-scales and maintenance costs for implementing a Portrait Foundation system.

Configuration that is published must be fixed. That is, the users taking the configuration must not be able to edit it. Otherwise the supplier of the configuration would not be able to support it. Configuration is published in packages. Each package exposes an interface, which is those parts of the configuration which are available for users of the package. Those parts which are not in the interface may be changed by the supplier in later releases of the package, but those parts which are in the interface cannot be.

Publishing configuration, that is, making an export of packages to another repository, is done in the Repository Manager tool. Refer to the *Repository Manager Guide* for a description of how to do this.

The interface on a package is defined using the Configuration Suite tool. This section describes how to do this.

Only users who are responsible for publishing configuration should define interfaces for packages. Packages which are not being published to other repositories should not have an interface defined.

5.2 Concepts

5.2.1 Interface properties

The interface for a package is defined at the configuration object level. The interface properties are properties of an object which control how it is exposed in the package interface.

The interface properties are as follows:

- **Public** – the object is visible outside of the package. Any object which is not declared Public will not be visible in any package which includes the current package.
- **Can Use** – narrows the scope of a Public object. It means that the object can be referenced by objects in other packages. If Can Use is declared False, then the owner of the package can freely modify it or remove it, since there cannot be any dependencies on it in configuration in other repositories.
- **Can Overlay** – narrows the scope of a Public object. It means that the object can be overlaid. Can Overlay can be set True when Can Use is False. This is useful for objects such as Model Class, where overlays are necessary but it might be undesirable to allow direct reference.
- **Can Extend** – narrows further the scope of Can Overlay. If set False, then the user of the package is unable to extend an overlay, although he may be able to override it.
- **Can Override** – narrows further the scope of Can Overlay. If set False, then the user of the package is unable to override an overlay, although he may be able to extend it.

5.2.2 Interface for included packages

When a package is included in another package, the inclusion has a type. The inclusion type is either Private or Public.

Public inclusion means that the topmost package inherits the interface of the included package. That is, the interface properties exposed in the topmost package for the included objects will be the same as those in the included package.

Private inclusion means that the topmost package does not expose the interface of the included package. That is, the interface properties for the included objects are all set to Private, whatever they were in the included package.

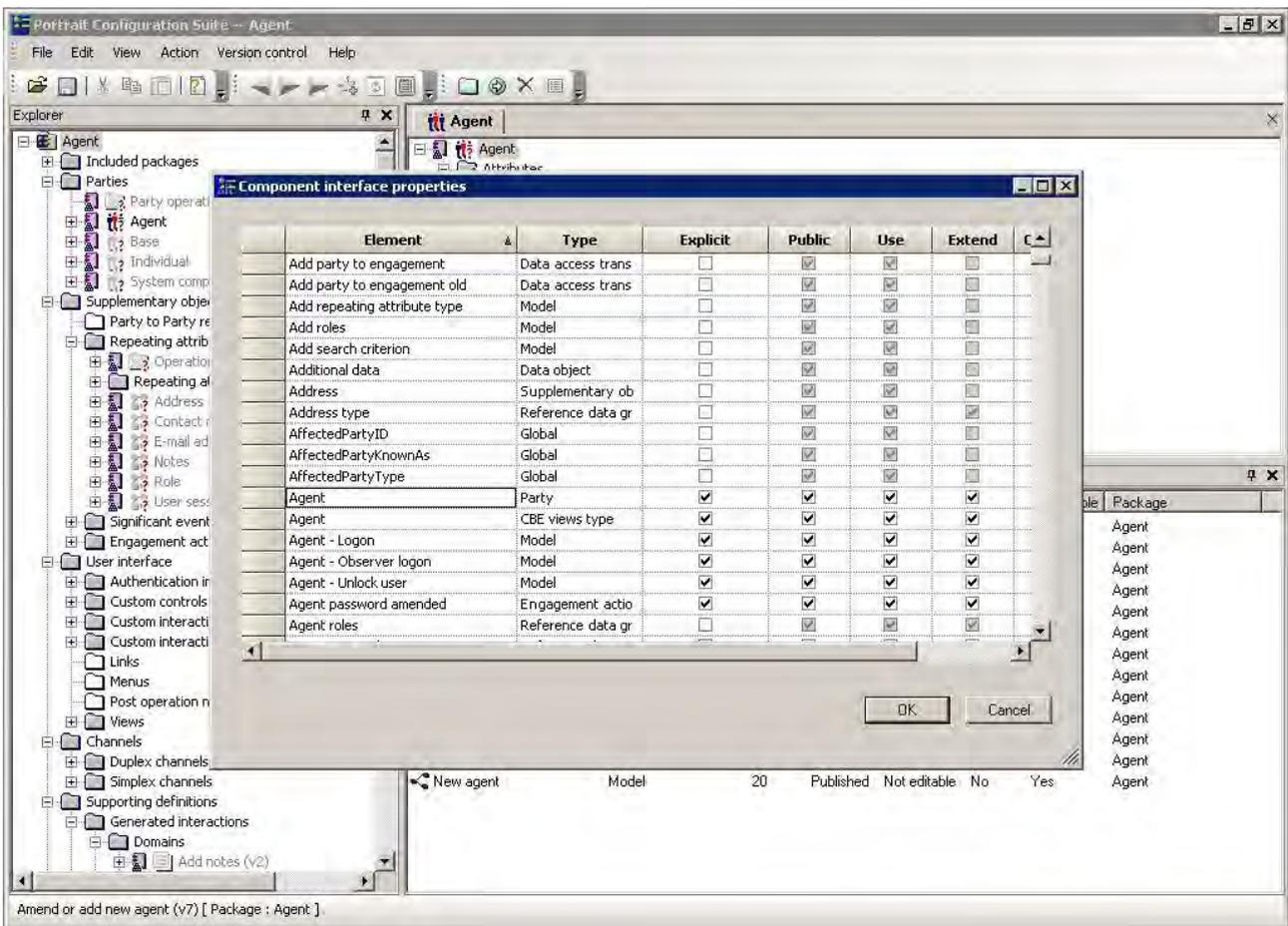
The interface properties which are inherited can be overwritten in the topmost package. This applies to both public inclusion and private inclusion. However, the scope of the interface can never be increased in the topmost package. For example, if the Can Overlay property for an object is set False in the included package, it cannot be changed to True in the topmost package.

5.3 User interface

5.3.1 Package interface properties dialog

To edit the interface on a package, open the package using **File Open package**. Select the package in the **Explorer** window. Select the option **Edit interface** from the context menu. This opens the **Package interface properties** dialog. This dialog displays the package interface as a list of the objects in the package, with the interface properties that have been set for each.

Figure 13 - Package interface properties



For objects which belong to the package, default values for the interface are set when they are created. These default values are displayed read only. To modify the default values, the user should check the Explicit check-box. This will enable

the check-boxes for the interface properties. Clearing the Explicit check-box will restore the interface to the default values.

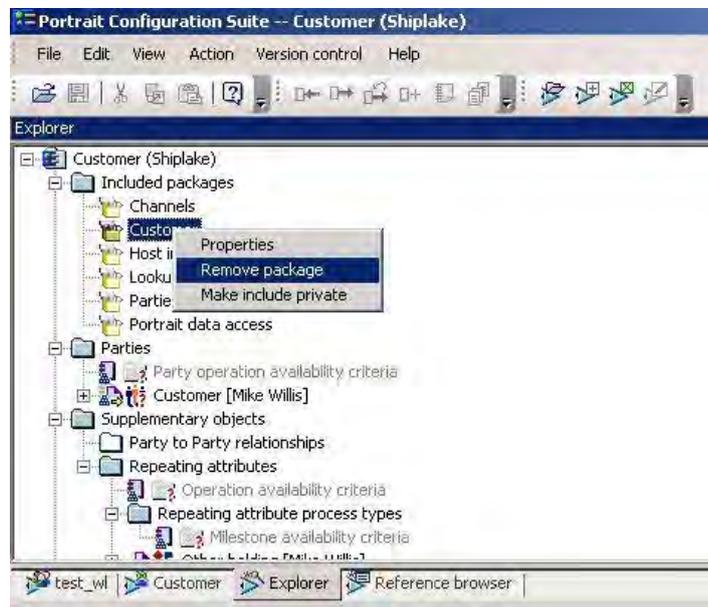
For included objects, the default interface is the package interface, qualified by the Private/Public Inclusion option. The user can modify the default interface in the same way, by using the Explicit check-box. Clearing the Explicit check-box will restore the default interface.

For included objects, changing the inclusion to private or public will lose any Explicit interface settings.

5.3.2 Make inclusion private/public options

To make the inclusion type of a package private, select the package in the **Included packages** folder. Select the option **Make include private**. This will re-set the included interface to make all objects private.

Figure 14 - Make include private menu option



Any values for the interface properties which were overwritten in the topmost package will be lost when the inclusion is made private.

If the inclusion type has been set private, it can be restored to a public inclusion type. Select the package in the **Included packages** folder. Select the option **Make inclusion public** in the context menu. This will restore the interface in the topmost package to the one inherited from the included package.

6 Working with overlays

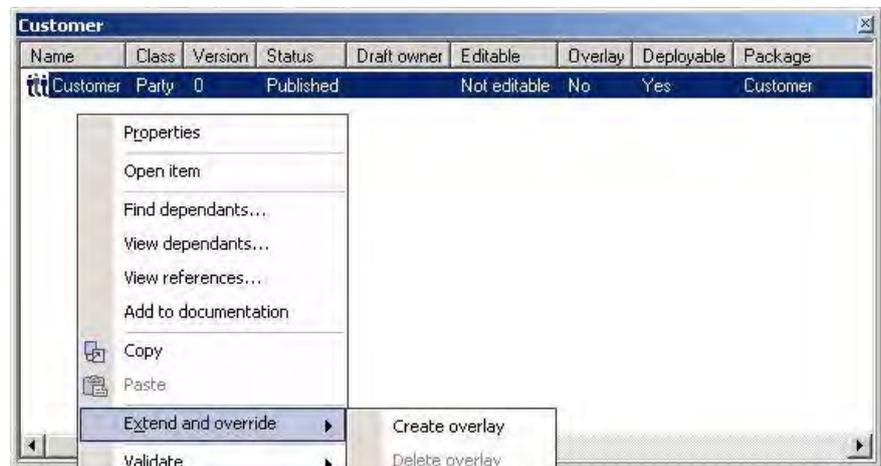
6.1 Introduction

This section tells you how to do the following.

- Create an overlay
- Extend an overlay
- Delete an overlay
- Create an override
- Edit an override
- Delete an override

6.2 Create an overlay

Figure 15 - Create overlay



To create an overlay, select the configuration object which is to be modified. Overlays can only be created on included objects.

Select **Extend and override** ► **Create overlay** from the Context Menu.

The overlay is now created. The existence of an overlay is indicated by the **Overlay** column in the work folder.. (See Figure 16.)

Figure 16 - Overlay created

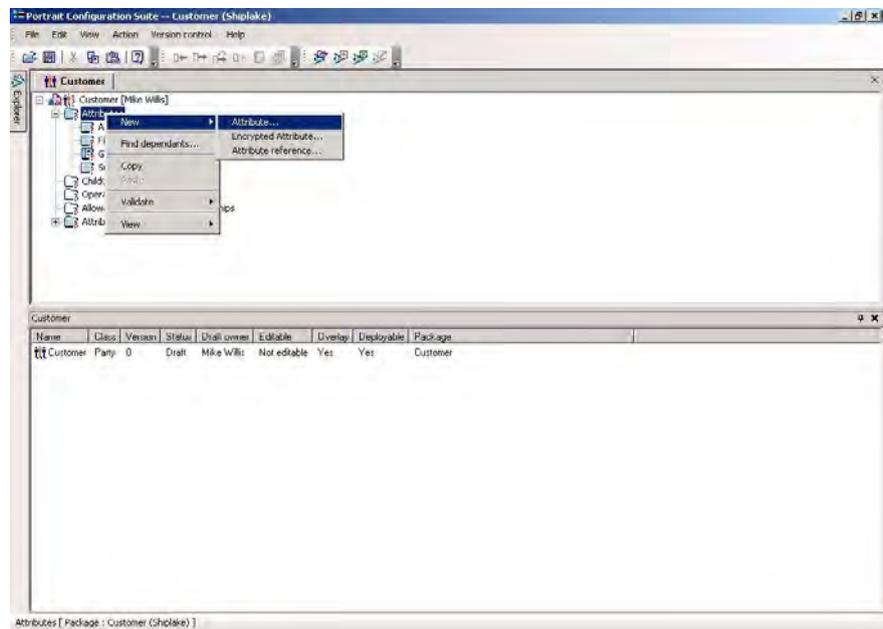


Before the overlay is created, the definition of the configuration object is read only, and is displayed in a greyed out format when opened. After the overlay is created, the object is displayed in normal font when opened. It can be edited as described in the following sections, by extending it, and by creating overrides. (Refer to the document *Portrait 3.1 Configuration Suite User Interface Enhancements*.)

6.3 Extend an overlay

With the overlay checked out, select a **New** option from within the configuration object. For example, in Figure 17 below, select **New attribute** or **New operation**. Complete the dialog. The new item will extend the party definition. The extensions exist only in the overlay, not in the original party definition.

Figure 17 - Extending an overlay

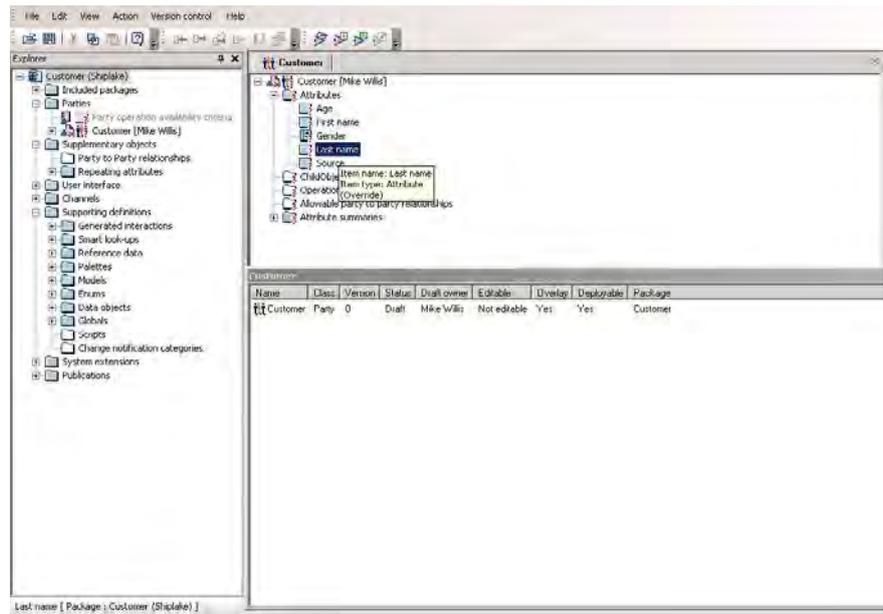


Extensions can be deleted or modified, by checking out the overlay and then editing it. Those parts of the party definition which belong to the underlying object will remain greyed out and cannot be modified.

6.4 Delete an overlay

With the overlay checked out, select **Extend and override** from the context menu. Select **Delete overlay**. The overlay is removed, together with all extensions and overrides which were created on it.

Figure 20 - Override example



The presence of an override can be detected using the tooltip, as illustrated above.

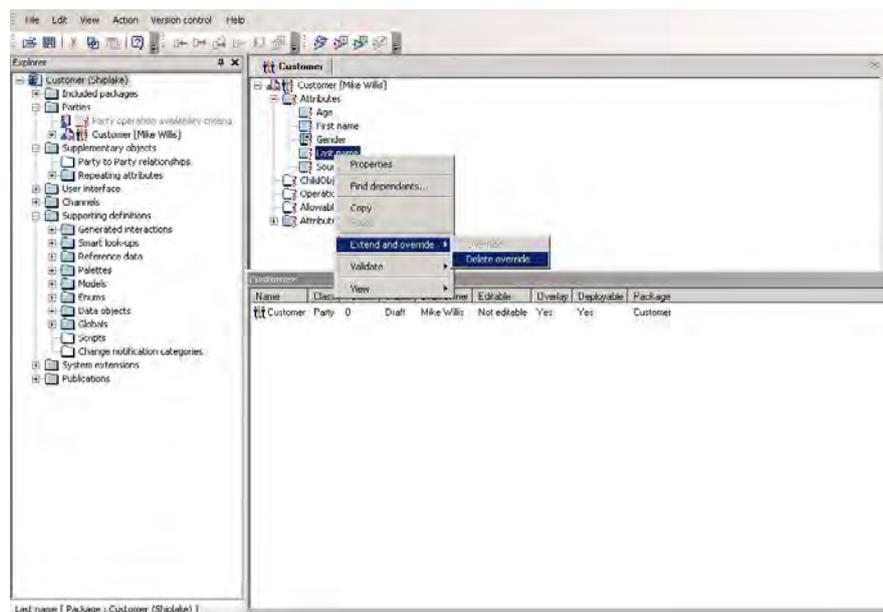
Not all properties can be overridden. Those properties which cannot be overridden will be disabled in the Properties dialog.

6.7 Delete an override

Select the item which has an override, and select **Delete override** from the **Extend and override** option in the context menu.

Deleting an override deletes the copy of the object's definition from the overlay. All changes which have been written to that copy are lost, and the original definition is restored.

Figure 21 - Delete override



Deletion of an override restores the properties to their original value. In this example, deletion of the “Last name” override will restore the attribute display name to “Surname”.

7 Reference browser

7.1 Introduction

The Reference Browser window enables the user to discover a configuration object's references. This is useful in fault-finding, and becomes essential when moving configuration from one package to another. (Refer to section 8.)

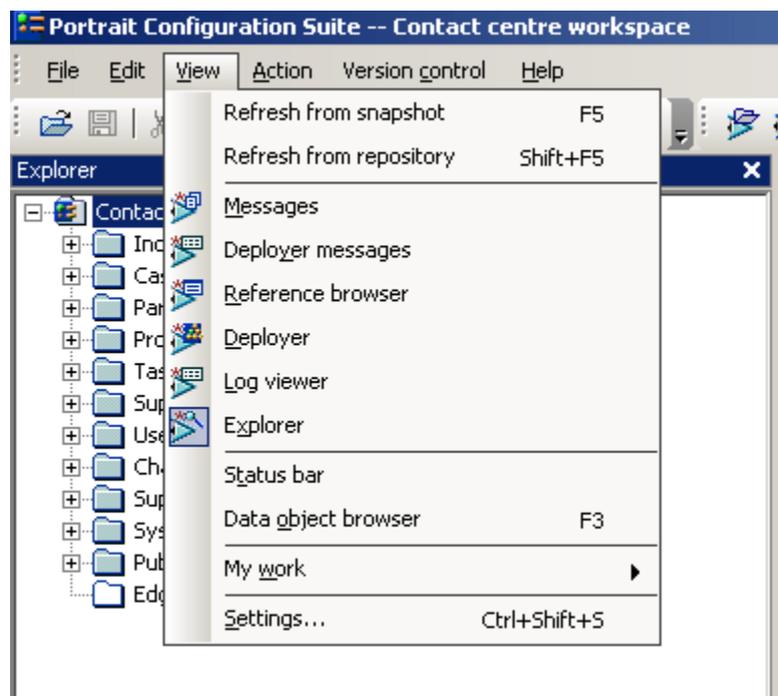
This section is for expert users. Most users of Configuration Suite will not need to use the Reference Browser.

7.2 Dialog overview

7.2.1 Opening the Reference Browser

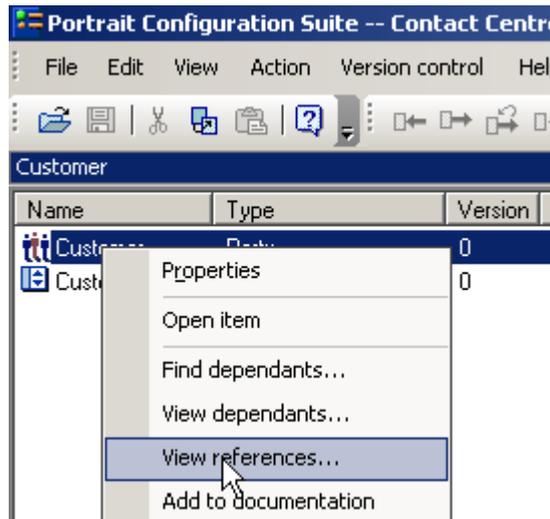
To open the Reference Browser, select **Reference Browser** from the **View** menu in the main menu bar.

Figure 22 -View / Reference browser menu item



To display references for a configuration object, select the object, open the context menu, and select **View references**. This will also open the window if it is not already open.

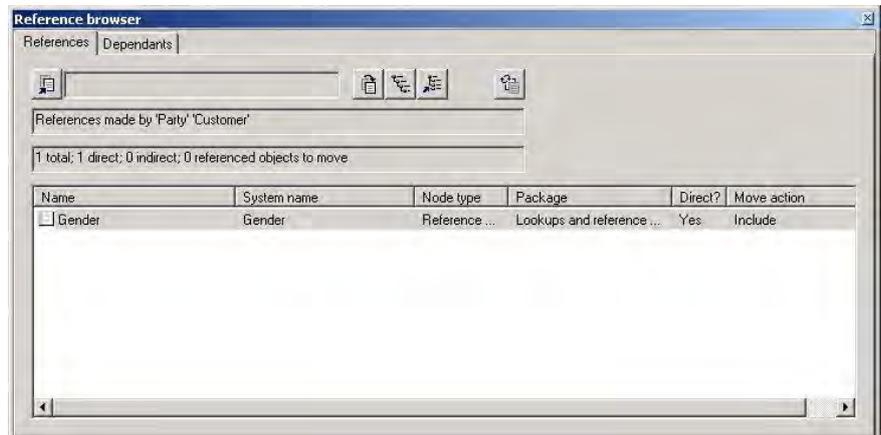
Figure 23 - View references context menu item



To display dependants for an object, select **View dependants** in the context menu.

7.3 Dialog layout

Figure 24 - Reference browser references tab



The dialog has the following parts.

- Tabbed folder - There are separate tabs for references and dependants. The layout is the same for both tabs.
- Toolbar - The toolbar supports the reference browser functions.
- Object - This field displays the name of the object.
- Count - The count field displays the number of references found.
- List - The list displays the references or dependants found.

7.3.1 Reference browser functions

The Reference Browser has the following functions.

- Select target package
- Move configuration
- View reference paths

- Locate in tree
- Refresh

7.4 User interface description

7.4.1 Toolbar

The toolbar has the following buttons.

	Select target package	This button displays the Select package dialog. See section 8.2.2.
	Move configuration	This button displays the Move configuration wizard . See section 8.2.2.
	View reference paths	This button displays the Reference paths dialog for the currently selected reference or dependant.
	Locate in tree	This button displays a highlight in Explorer View for the object which is pointed by the currently selected reference or dependant.
	Refresh	This button refreshes the list in the references and the dependants tabs.

7.4.2 Object

The object field displays the object for which references or dependants are being displayed. The text is in the form "References made by [object type] [object name]", as in the example "References made by Party Customer".

7.4.3 Count

The Count field displays the following counts.

- the total number of references or dependants in the list
- the number of objects in the list with a direct relationship to the object selected in the configuration tree
- the number of objects in the list with an indirect relationship to the object selected in the configuration tree
- the number of objects whose definition can be moved out of the current package into another package

7.4.4 List

The List displays the list of referenced or dependent configuration objects.

- **Name** The display name
- **System name** The system name
- **Node type** The object type
- **Package** The package containing the object
- **Direct** The type of the reference

The type of a reference is one of:

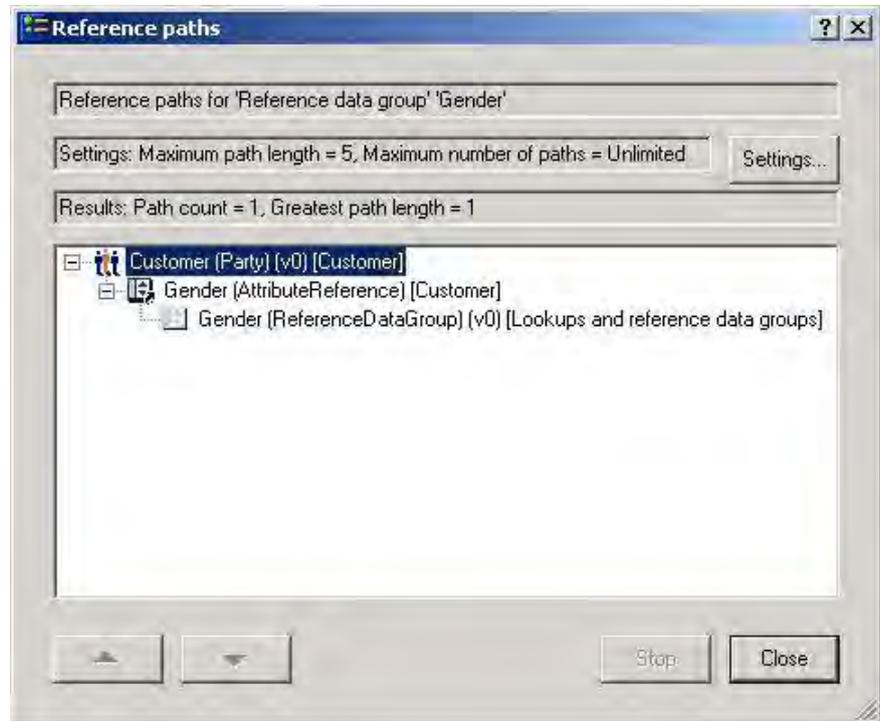
- **Direct (Yes)** The reference is direct.
- **Indirect (No)** The reference is indirect, through other configuration objects.

7.5 Reference paths dialog

7.5.1 Overview

The Reference Paths dialog is used to display the path by which one object refers to another object.

Figure 25 - Reference paths dialog



The path is displayed as a tree view. The root node in the tree is the source of the reference. The leaf node in the tree is the destination of the reference.

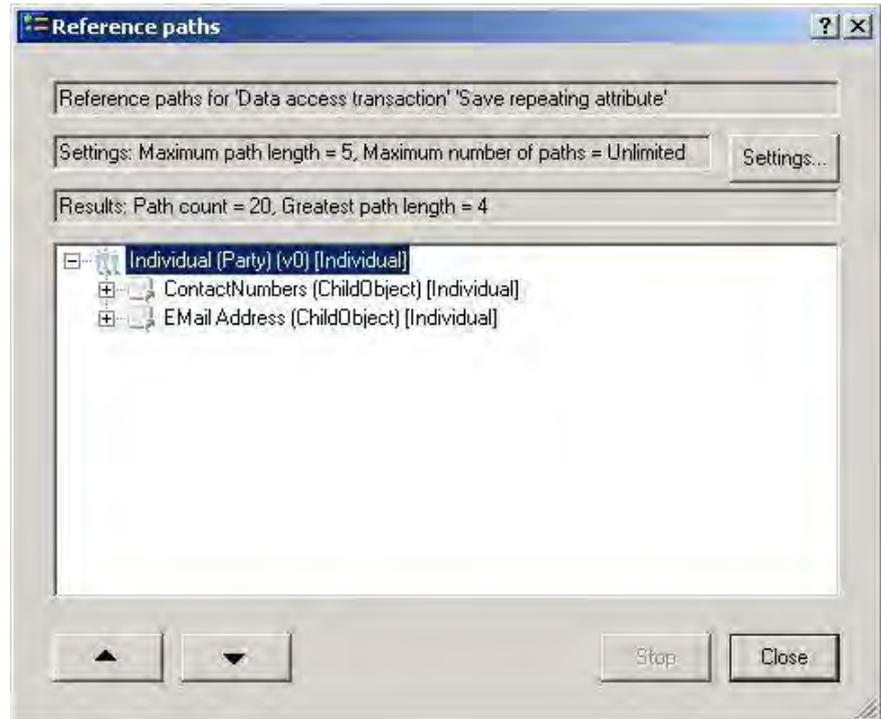
In the above example, the **Reference browser** has been opened from the object Party Customer. In the list of references, the object Reference Data Group Gender has been selected. The **Reference paths** dialog shows that Customer has an attribute Gender which uses the Reference Data Group Gender.

If this example was opened from the **Dependants tab**, the display would be the same. The difference would be that in the **References tab** the object selected was Reference Data Group Gender. In the **Dependants tab** the object selected was Party Customer.

The path length of a reference path is the number of steps from the source to the destination. In this example the path length is 2.

There can be multiple paths between two objects.

Figure 26 - Multiple reference paths



In this example the data object Party Individual references the data access transaction Save Repeating Attribute through two main paths, one for saving contact numbers and one for saving e-mail address. More subsidiary paths would be discovered as these paths were expanded.

If paths lengths are large, or there are many paths, the time taken to search for reference paths can be long. The user can stop search using the **Stop** button. He can also restrict the search using the **Settings** button.

7.5.2 Fields

The dialog has the following fields.

- **Destination** The Destination label displays the object which is being referenced.
- **Settings** Settings displays the current settings. The settings are the **Maximum path length** and the **Maximum number of paths**.
- **Results** Results displays the number of paths, and the length of the greatest path length.
- **Path list** Path list displays the paths that were found, organised in a tree view. A path consists of the items in the configuration definitions through which objects are linked. For each configuration item in the path, the following information is displayed.
 - Item type
 - Item name
 - Reference descriptor, such as Child Object, Model Reference
 - Object type for the object which is referenced

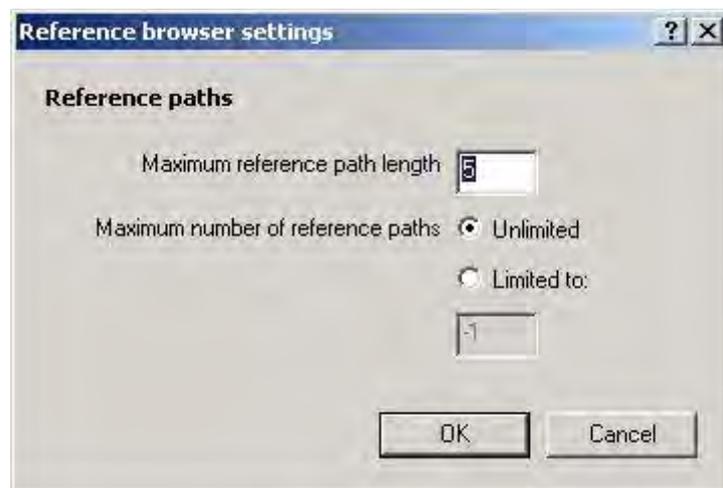
7.5.3 Buttons

The Reference paths dialog has the following buttons.

	Settings	Displays the Settings dialog.
	Next	Moves the selection for the reference paths to the next configuration object in the Reference browser list.
	Previous	Moves the selection for the reference paths to the previous configuration object in the Reference browser list.
	Stop	Stops a search.
	Close	Closes the dialog.

7.6 Reference browser settings dialog

Figure 27 - Reference browser settings dialog



The **Reference browser settings** dialog provides two settings to control the depth of search for reference paths.

- Maximum reference path length
- Maximum number of reference paths

Maximum reference path length

The default value of the Maximum reference path length setting is five. If you find that the Reference paths dialog fails to display a reference path that you suspect exists, it may be that the reference path length is greater than the Maximum reference path length setting., you can increase the value of this setting.

Maximum number of reference paths

The default value of the Maximum number of reference paths setting is Unlimited. This means that the Configuration Suite will attempt to evaluate all reference paths (subject to the maximum reference path length setting). If you find that the evaluation of reference paths takes longer than you wish to wait, you can limit the evaluation to a maximum number of reference paths. The Configuration Suite will then stop evaluating reference paths when it has identified the specified number of reference paths.

8 Move configuration wizard

8.1 Introduction

There are a number of circumstances when it is desirable to move configuration from one package to another. The following are examples.

- When developing new packages it may become apparent that a greater level of granularity is required and the configuration in the new package should be split into a number of smaller, lower level/better focused packages.
- A workspace developed prior to Release 3.0 needs to be broken down into packages containing just the project specific configuration (where possible) when it is being upgraded to 3.0. and the new Foundation and Contact Centre packages are intended to be used.
- When configuration is added to the wrong package and it needs to be moved.

The **Move configuration wizard** enables the user to move a configuration object from one package to another. Other objects which are referenced by it are moved at the same time.

Because the user must be able to see an object's references before moving it, the **Move configuration wizard** is accessed from within the **Reference browser window**. (See section 7.)

8.2 Moving configuration

Moving configuration means moving an object from a source package to a target package.

When moving an object, the object's references must be moved with it, to maintain the integrity of the configuration. Where the referenced objects are part of the source package, they are moved as well. Where the references are to included packages, those packages will be included in the target package. Optionally, the referenced packages can be removed from the source package, if they are no longer needed.

8.2.1 Before moving

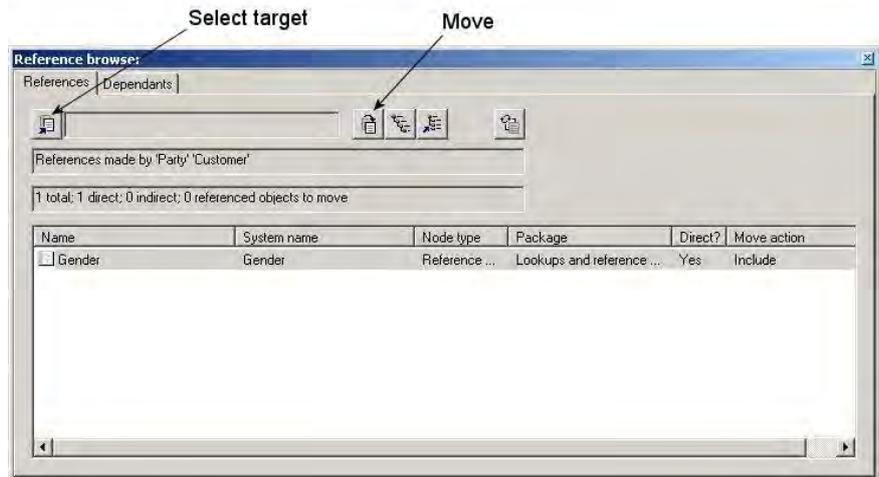
Before moving an object, check the following.

- 1 Check that the source package is the current package. Since the move will change the source package, it must be the current package.
- 2 Check that the target package is included in the source package. A move can only be made to an included package. This ensures the integrity of the source package after the move.
- 3 Check that the object to be moved is published. The move will be stopped if any of the objects being moved are in a checked out or draft state.
- 4 Check that no other user has the object checked out, or any of its direct references.
- 5 If any of the objects being moved is an overlay, check that there is no overlay on the same original in the target package. Overlays can be moved, like an object, but the move will be stopped if there is a corresponding overlay in the target.

8.2.2 Using the Move configuration wizard

The Move configuration wizard is accessed in the Reference browser.

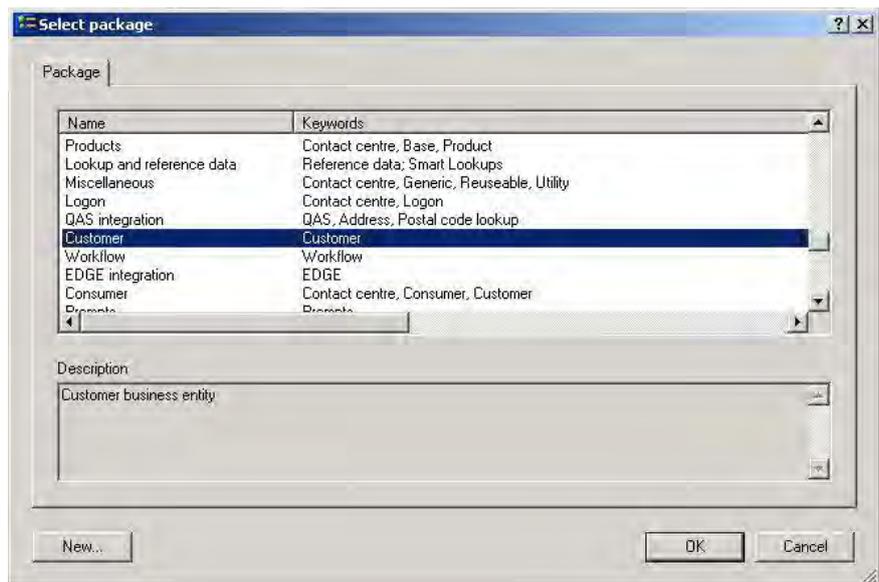
Figure 28 - Reference browser



To use the wizard, perform the following steps.

Press the **Select target** button. This displays the **Select package** dialog. Select the target package from the list.

Figure 29 - Select package



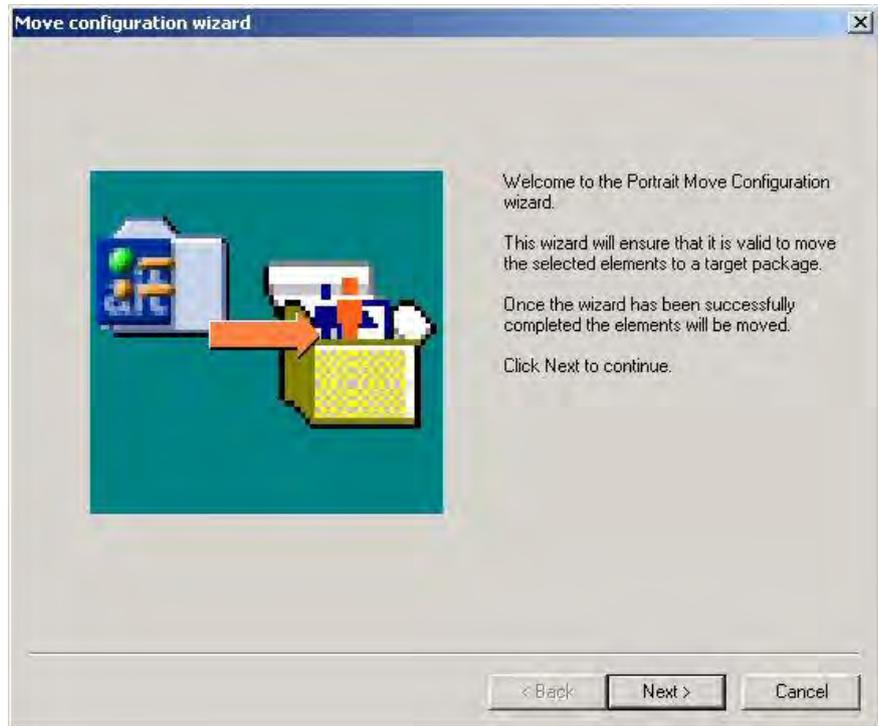
The target package is displayed in the **Reference browser**.

Figure 30 - Reference browser



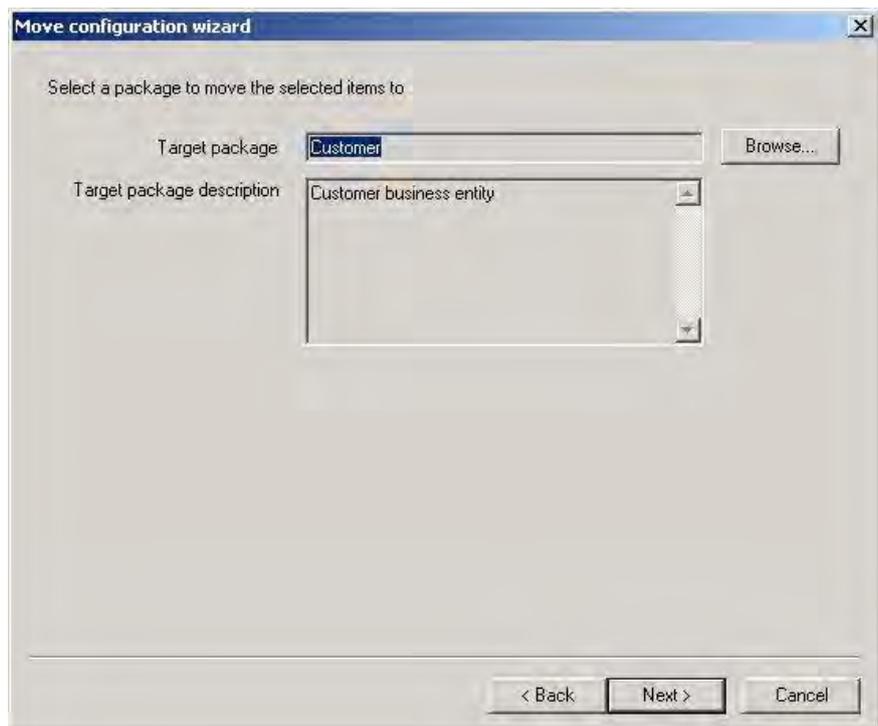
Press the **Move** button. The start page of the Move Configuration Wizard is displayed.

Figure 31 - Move configuration wizard start page



Press **Next**. The target package page is displayed.

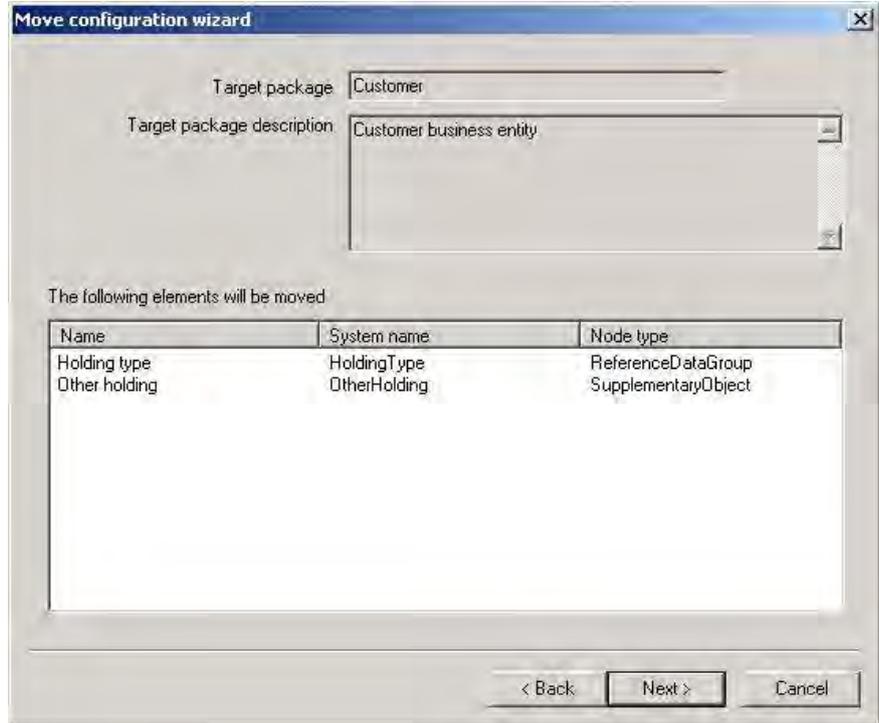
Figure 32 - Move configuration wizard target package page



By default, the page displays the package selected in the **Reference browser**. The target can be changed by using the Browse button.

Press **Next**. The objects list page is displayed. These are the objects which will be moved from the source package to the target package.

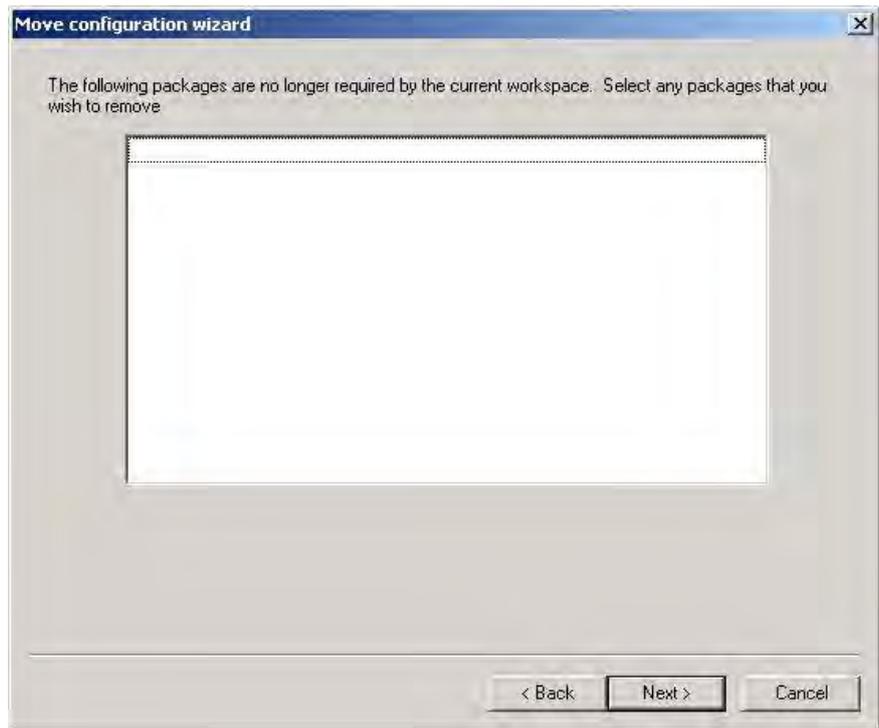
Figure 33 - Move configuration wizard objects list page



In this example, the object **Other holding** is the current object in the Reference Browser. The reference data group **Holding type** is part of the source package, and is referenced by **Other holding**, so it will be moved with it.

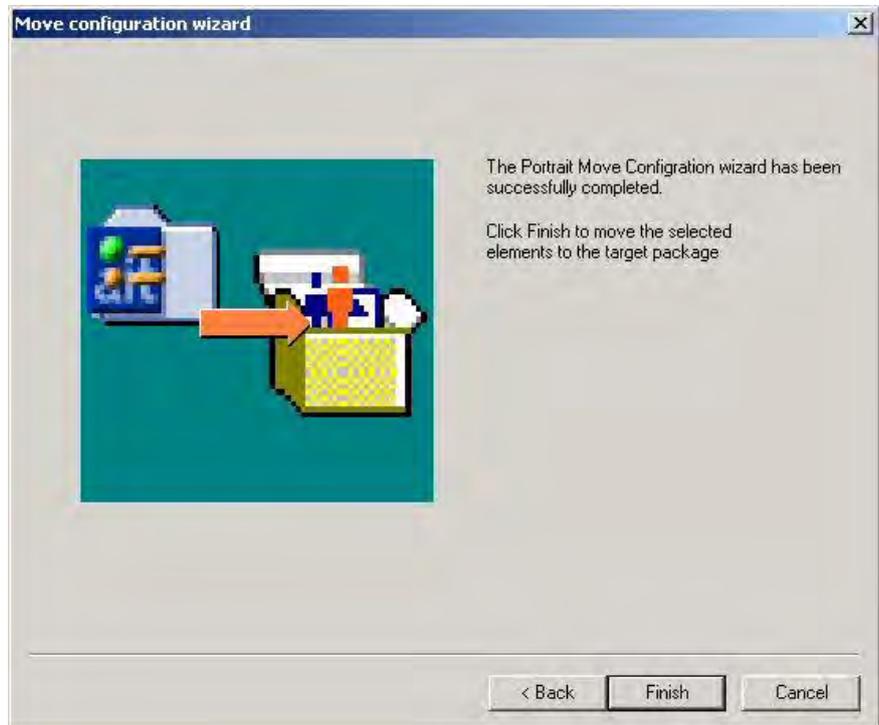
Press **Next**. The exclusions page is displayed. This page lists packages which are no longer required in the source package because there will be no direct references to them after the move. Normally the user will select all the packages listed. Optionally he may choose to keep some if he intends to use them later.

Figure 34 - Move configuration wizard inclusions page



Press **Next**. The confirmation page is displayed.

Figure 35 - Move configuration wizard confirmation page



The user should press **Finish** at this point to confirm the move. If he does not wish to complete the move he should press **Cancel**.

8.2.3 After moving

After the move has been completed, the following changes have been made.

- 1 The selected object, and any objects referenced by it in the source package, have been moved to the target package.
- 2 The target package now includes any packages which are referenced by the objects which have been moved.
- 3 Those packages which were selected on the exclusions page have been removed from the source package.