



Using the Contact Centre Workspace - Adding a party package

Edition 1.1

10 January 2013





Portrait Foundation Using the Contact Centre Workspace - Adding a party package

©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.co.uk/software/)

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

This document is intended to provide information that will allow a Portrait configurator to implement a solution, based on the *Contact centre workspace*.

The *Contact centre workspace* provides functionality that will allow the user to log on to a contact centre application and attempt to take a call.

This document does not provide the information necessary to *build* the solution components or create the appropriate install.

However, the build process must be performed before it is possible to run the configuration described in this document, and is described in the Foundation SDK user guide.

Intended audience

This document is intended to be used by configurers and developers involved in creating a Portrait serviced application e.g. branch, contact centre.

Related documents

Software Development Kit User Guide (*Foundation SDK User Guide*)

Using the Contact Centre Workspace Overview

Software release

Portrait Foundation 4.2 SP1 or later.

Contents

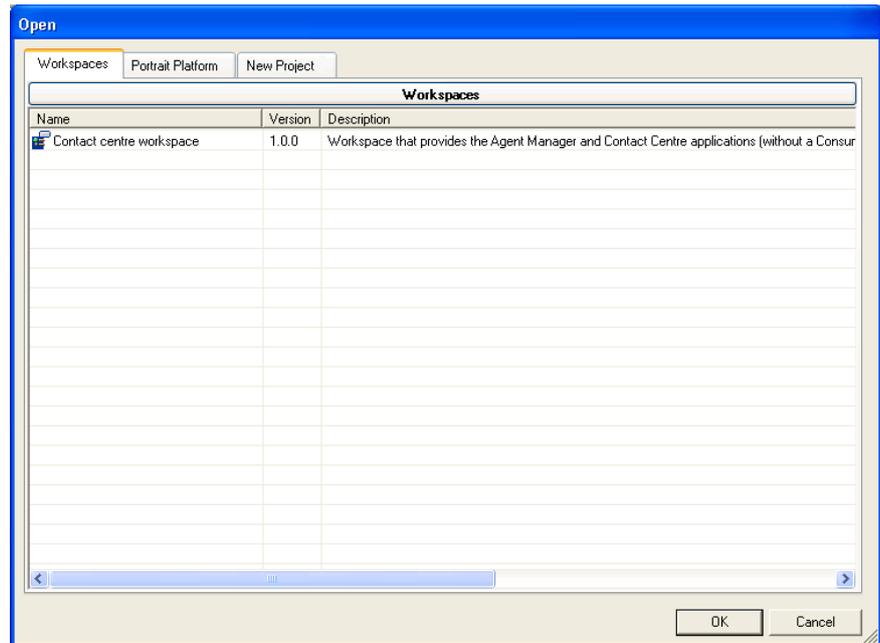
1	Introduction	6
1.1	The Contact centre workspace	6
1.2	Functional overview	6
2	Using this document	7
2.1	Checklist	7
2.2	Indication of mandatory/optional steps	7
2.3	Nomenclature	7
2.4	Starting point	7
3	Adding a party package	8
3.1	Include package	8
3.2	Create the party overlay	9
3.3	Modify contact centre sidebar menu	11
3.4	Make party type searchable	12
3.5	Add class members	13
4	Add CBE views	20
5	Controlling party search	22
5.1	Party attribute search criteria and results view	22
5.2	Repeating attribute search criteria	24
5.3	Contract attribute search criteria	26
6	Publish and Deploy	28
	Appendix A Integration Checklist	29
A.1	Mandatory steps	29
A.2	Optional steps	29

1 Introduction

1.1 The Contact centre workspace

This document is based on the premise that your implementation will evolve from a primitive workspace that is provided as part of a standard Portrait installation.

This is called the *Contact centre workspace* and it will appear on the workspaces tab in the repository browser when your repository has been created.



Unlike packages, workspaces are **all owned** by your project/implementation. In this context, *ownership* determines whether you can modify the content of configuration objects in addition to extending and overriding them.

1.2 Functional overview

The assumption is that your starting point was the *Contact centre workspace*. This workspace includes the *Agent* configurable business entity (CBE) definition along with the objects from which it inherits e.g. *Individual*. Generally, implementations will extend this party definition via their own party packages as defined by their individual business needs.

However, a transferable *Consumer* package is provided for use by an implementation and this is employed as an example in this document to illustrate how party definitions can be extended and related party functionality can be utilised.

2 Using this document

2.1 Checklist

You should print the Checklist provided in Appendix A - Integration Checklist.

This will provide you with a summary of all the high levels steps you need to consider in order to integrate a party definition into your workspace.

The checklist contains a *Done* column so that you can check off the items completed.

2.2 Indication of mandatory/optional steps

The checklist provides a summary of mandatory steps and optional steps. Clearly, you must complete the mandatory steps before you can use the packaged functionality.

You should review the optional steps carefully in case they provide the opportunity for a better fit with your implementation.

2.3 Nomenclature

The document is slanted towards party being a human entity whose details are being maintained and who is being supplied services and/or products by an organisation across channels. However, party definitions and some supplied processes have also been utilised where the party is not a human entity e.g. an emergency services incident, an organisation.

2.4 Starting point

This document assumes that you have already opened your workspace and that your workspace is based on or is the *Contact centre workspace*.

3 Adding a party package

The *Contact centre workspace* is intended to provide the functionality that allows *Agents* to handle requests for service or purchases by other parties. The workspace already includes a party definition for the *Agent* party; however it does not include any definitions for other parties. It is expected that you will add these definitions yourself.

You can do this by including a party definition that has been supplied to you or by creating your own party definition.

This section describes the parts of the workspace that should be extended in order to integrate your definition(s) into the functionality that is already provided.

Until you provide at least one definition, you will not be able to exercise the full capabilities of the contact centre.

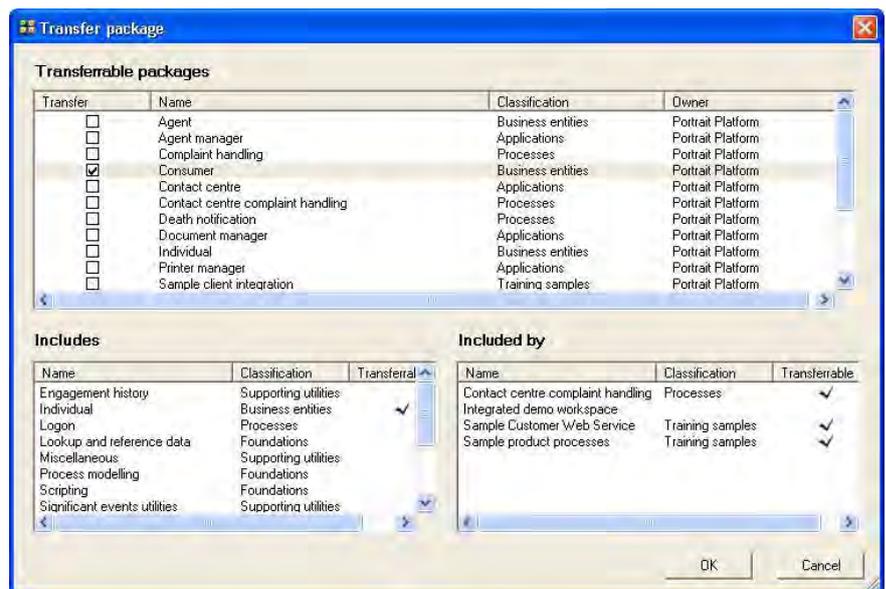
3.1 Include package

You can create your own party definition or use a party definition provided by another supplier. Typically, party definitions provided by other suppliers (e.g. Portrait Foundation) will be transferable. You can use transferable packages in the same way as any ordinary package or you can choose to transfer the package. If you transfer the package then:

- You will own all the configuration provided by the package
- In addition to extending the configuration, you will be able to edit it.
- You will NOT be able to take any upgrades to the package

To transfer a supplied party package:

- 1 Open the Portrait Repository Manager
- 2 From the **Actions → Configuration** menu, select the **Transfer package...** menu item.

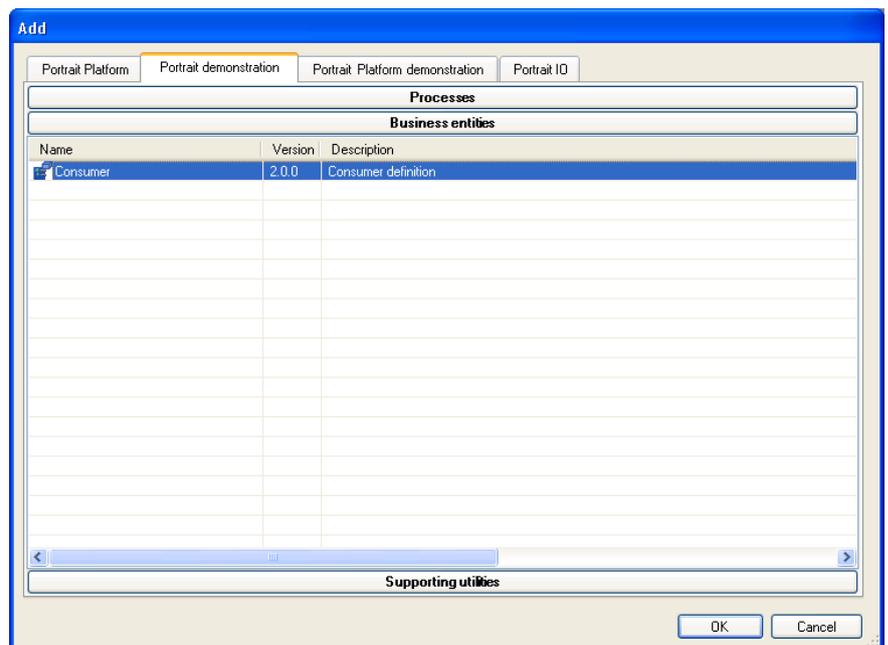


- 3 The *Transfer package* dialog will be displayed. Select the package that you wish to transfer. The act of transferring this package may also incorporate some included packages.
- 4 Click on the **OK** button.

This will create the package which you can then add to the workspace.

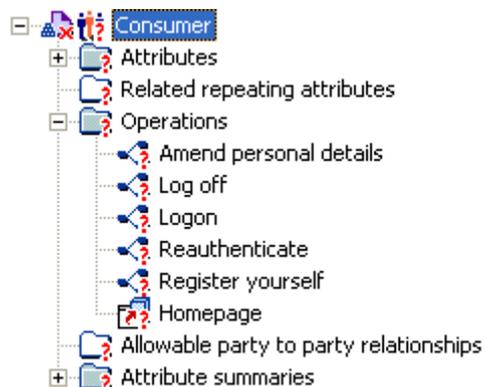
To include this package or any other party definition package that you intend to use:

- 1 Select the **Included packages** folder at the top of the configuration suite *Explorer* and from the context menu select **Add package...**
- 2 Select the **relevant owner's tab** in the repository browser. If you have created or transferred a package this will be the tab with your project owner as the tab name.
- 3 Navigate to the **Business entities** shutter within the tab (Party definition packages are usually classified as business entities).
- 4 Select the package and click **OK**.

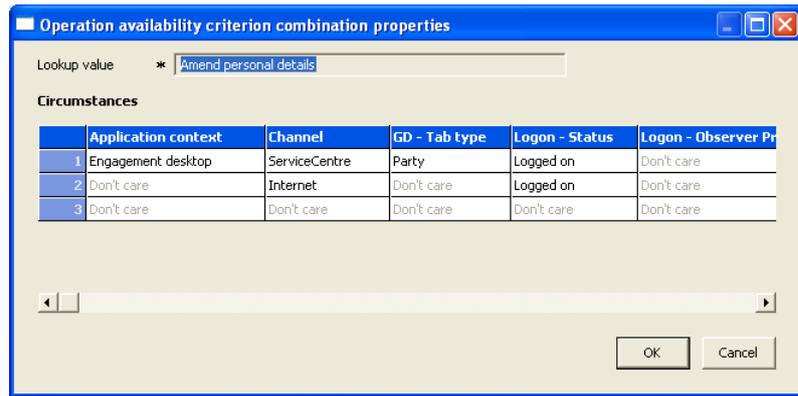


3.2 Create the party overlay

- 1 Once the package is included, select the party definition in the workspace and then using the context menu select **Extend and override → Create overlay**



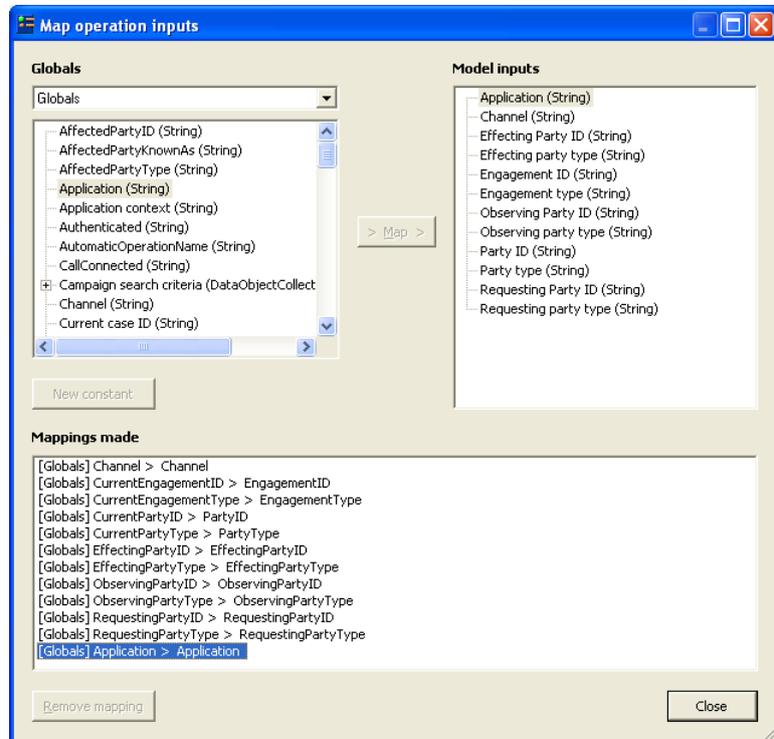
- 2 For each of the operations, set the operation availability by selecting the **Set operation availability...** menu item from the context menu on the operation configuration item.



Bear in mind that you may need to make the operations available on multiple channels. The criteria will be determined by the configuration in your workspace. However, the criterion list will typically include:

Application context	Application contexts are usually set by tabs and/or operations in the Generic desktop. They are used to specify the second layer of state information (Application is usually the first). Typically when you log on to the Contact centre, the initial context is 'Agent desktop'. Once a call has been taken, the context is usually 'Engagement desktop'. After wrap-up, the context usually reverts to 'Agent desktop'. Finally, as a result of either pick work or get work, the context is usually 'Workflow desktop'. You can create new context values by adding items to the Application contexts reference data group. You can set the context by setting the Application context global as a result of an operation outcome MUM or by setting one of the variables on the appropriate tab in the first (invisible) row of tabs. The value for the variable location ID should be 'ApplicationContext'.
Channel	Indicates which Channel the operation should be available for. Typically this will be 'ServiceCentre' and/or 'Internet'.
GD - Tab type	This value comes from a global called 'CurrentPrimaryTabType'. The global is usually set by tabs on the primary (first visible) row of tabs. Values typically include 'Party', 'Task' and 'Case'. This criterion should be used if you want to ensure that the operation is only available when a Party tab is currently selected.
Logon - Status	Use the criterion if you want to ensure that the operation is only available when the user is logged on.
Logon - Observer present	Use the criterion if you want to ensure that the operation is only available when an observer is logged on.
Telephony - Agent state	Use this criterion if you want to ensure that the operation is only available in certain telephony states. Unless you are certain that CTI is going to be implemented for your implementation it is probably better not rely on this criterion (the criterion is present to control the availability of agent operations in a CTI environment).
Telephony - Outbound call state	See Telephony - Agent state above.

- It is likely that the party definition that you are including will already have operation input and output mappings declared. It is possible that these will be incomplete if your workspace has additional globals that you believe are relevant to or affected by any of your party operations (e.g. **Application** → **Application**). In this case you will need to add the appropriate MUMs by selecting **Map model inputs...** and/or **Map model outcomes/outputs...** from the operation context menu.

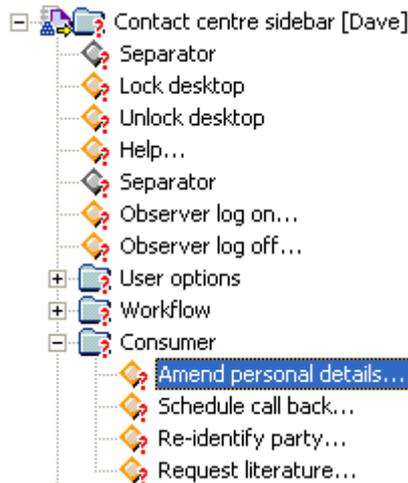


- Validate.
- Add the overlay to source control and publish.

3.3 Modify contact centre sidebar menu

The operations provided by your included party type must be added to the menu displayed in the contact centre before they will become visible to the user.

- Check out the **Contact centre sidebar** menu (in the User interface/Menus folder)
- Create a new sub-menu in the sidebar. Usually this will take the name of the party type that you have added. You can, however, give the sub-menu any name. You can also skip this step completely if you do not think that a sub-menu is appropriate.
- Copy the relevant operations from the party definitions (some operations may not apply to the Contact centre).
- Select the new sub-menu item created above and select **Paste as→Operation menu item** from the context menu.



- 5 At this point you may want to add other operations that apply to the party, but which are provided by other packages (e.g. **Schedule call back...**).
- 6 Validate.
- 7 Check the menu into source control and publish.

3.4 Make party type searchable

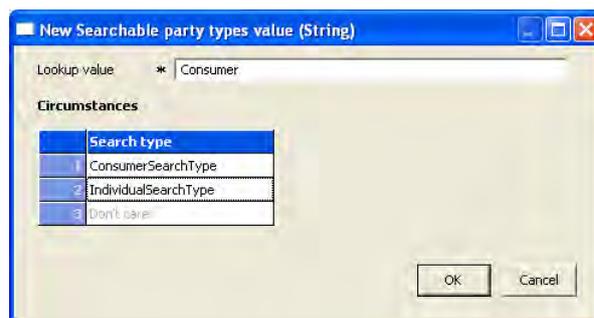
The **Search for party** custom interaction (CI) used by the contact centre when identifying a party (e.g. during **Take call**) uses a smart lookup to work out which party types can be displayed in the Party type combo box in different circumstances.

The criterion for selection is called **Search type** and there are a number of values that are already used in various packages to identify which party types to display. These are:

ConsumerSearchType	Used wherever the CI is intended to search for Contact centre consumer. The word is used here in its general form. It does not refer to a specific party type.
IndividualSearchType	Used whenever the CI is intended to search for any party type that represents a person (as opposed to a corporation or a system). Usually Agent and a 'Consumer' are both available when this search type is used.

If you use the **Search for party** CI directly you can supply any value corresponding to the criterion value you have setup in the smart lookup.

- 1 Create an overlay on the smart lookup called **Searchable party types**.
- 2 Add a new value where the value is the system name of your party type.



- 3 Validate.

- 4 Add the overlay to source control and publish.

3.5 Add class members

The Contact centre workspace is provided with a number of model class definitions that should be used to implement party specific behaviour within the generic framework. This section provides a brief description of the model classes and list the steps required to add party specific class members.

In general, it will only be necessary to add a class member if you intend to use the related functionality (i.e. there are optional parts of the framework).

3.5.1 Schedule call back - Is call appropriate

The model class is used in the schedule call back process to test that a scheduled call back is appropriate for a given party type. For example, members can test whether or not the party is deceased or if a 'Do not call' flag has been set.

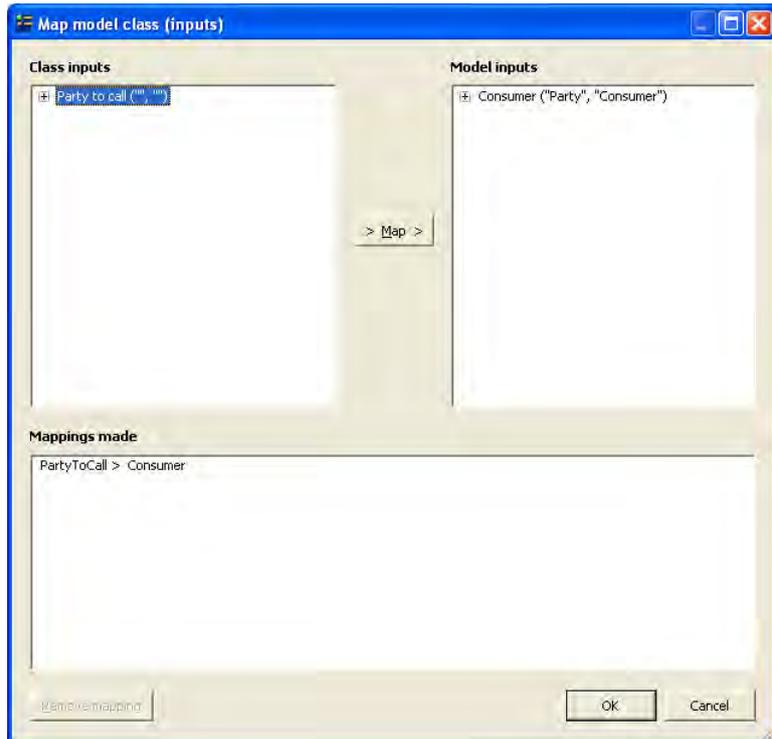
The model class outcomes are self explanatory (**Call** and **Do not call**). In the **schedule call back context**, the 'Not a member' outcome will allow the call scheduling to proceed.

The name of the class member MUST be the same as the party type. If the call should not be scheduled, class members should also display reasons why it should not be scheduled.

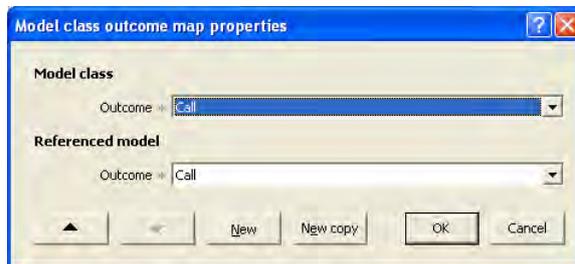
- 1 Create an overlay on the Schedule callback - Is call appropriate model class.
- 2 Add a class member (system name MUST be the same as the system name of the party) referring to a business operation model that provides the **appropriate functionality**. Remember, if you don't want to test for the appropriateness of the call, then DO NOT ADD a member and ignore the following steps.



- 3 Map the class member inputs (in this case, the inputs to the model should be the party data object) by selecting **Map model class inputs...** from the class member context menu.



- 4 Create model class outcome mappings for each of the outcomes (there are no outputs to map) by selecting **Create model class outcome mapping...** from model class outcome maps context menu.



- 5 Validate.
- 6 Add the overlay to source control and publish.

3.5.2 Death notification - Get deceased indicator

The model class is used to extract the values required by the death notification process to determine whether a party is deceased (deceased indicator, date of death & date of birth). The values are determined from a party object passed in.

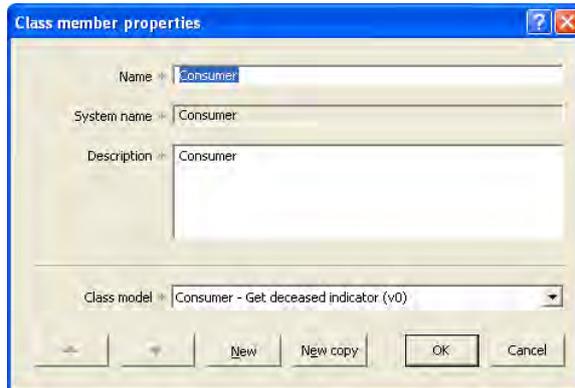
The class member is responsible for extracting the output values from the party data object using a combination of mapping and processing logic.

The system name of the class member **MUST** be the same as the system name of the party type.

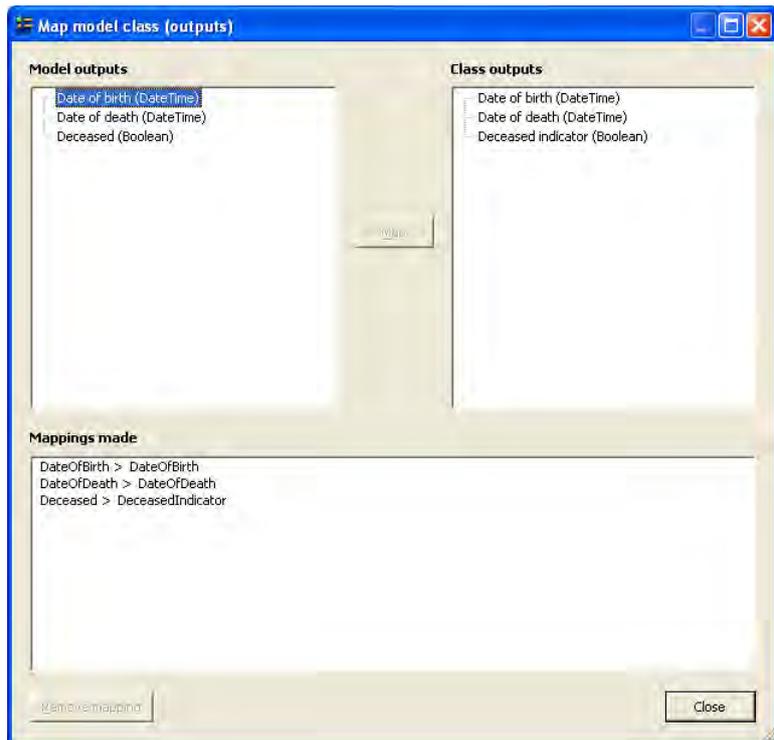
The Model class supports an 'OK' and a 'FAIL' outcome. In the death notification context, the FAIL and the 'Not a member' outcomes will cause the process to assume that the party is NOT deceased.

- 1 Create an overlay on the Death notification - Get deceased indicator model class.
- 2 Add a class member (system name **MUST** be the same as the system name of the party) referring to a business operation model that provides the **appropriate functionality. Remember, if you don't want to test for the party**

being deceased, then DO NOT ADD a member and ignore the following steps.



- 3 Map the class member inputs (in this case, the inputs to the model should be the party data object) by selecting **Map model class inputs...** from the class member context menu.
- 4 Create model class outcome mappings for each of the outcomes by selecting **Create model class outcome mapping...** from model class outcome maps context menu.
- 5 Map the member outputs for the OK outcome by selecting **Map model class outputs...** from the outcome map context menu.



- 6 Validate.
- 7 Add the overlay to source control and publish.

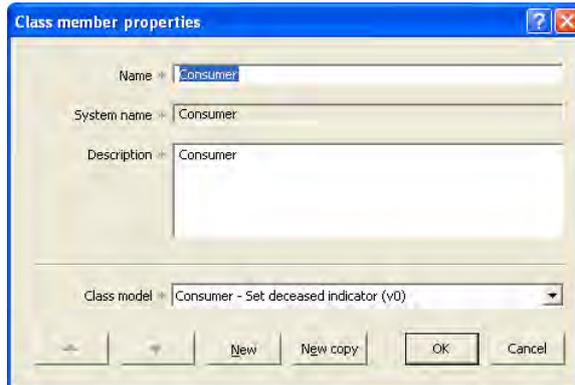
3.5.3 Death notification - Set deceased indicator

This model class is used to set the attribute(s) that will be used for a specific party type to indicate that the party is deceased.

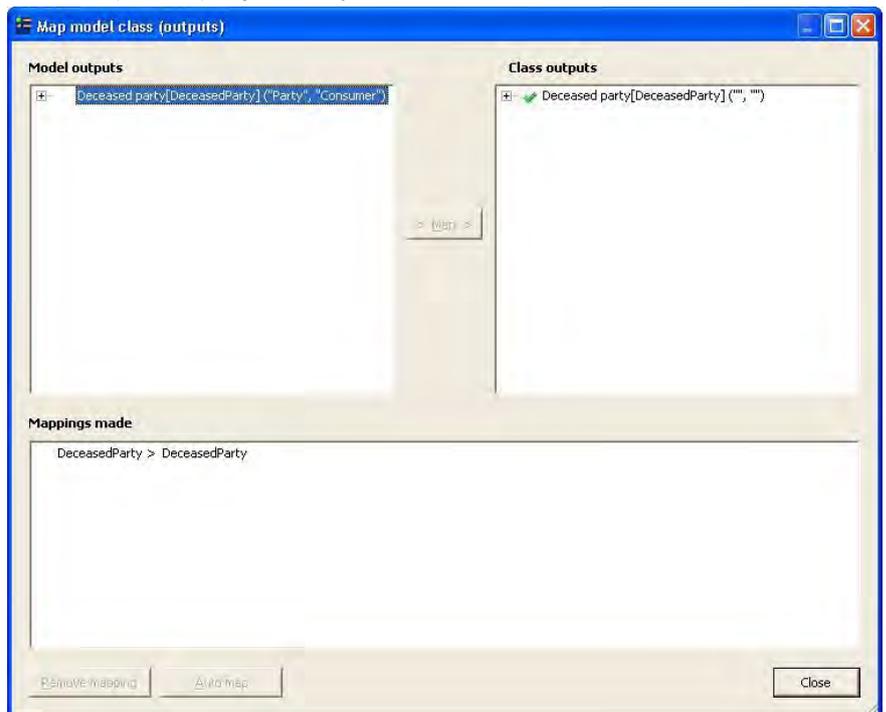
The system name of the class member MUST be the same as the system name of the party type. Class members need not save the party. The party will be saved by the **Death notification** process.

The Model class supports an 'OK' and a 'Fail' outcome. In the death notification context, the Fail and the 'Not a member' outcomes will cause the process to fail. It is imperative, therefore, that you supply a working member if you intend to use the Death notification process.

- 1 Create an overlay on the Death notification - Set deceased indicator model class.
- 2 Add a class member (system name MUST be the same as the system name of the party) referring to a business operation model that provides the appropriate functionality. Remember, you must add a working member if you intend to use the Death notification process



- 3 Map the class member inputs (in this case, the inputs to the model should be the party data object, a Boolean to indicate if the party is deceased and a date of death) by selecting **Map model class inputs...** from the class member context menu.
- 4 Create model class outcome mappings for each of the outcomes by selecting **Create model class outcome mapping...** from model class outcome maps context menu.
- 5 Map the member outputs for the OK outcome by selecting **Map model class outputs...** from the outcome map context menu (the member model should output the party data object).



- 6 Validate.
- 7 Add the overlay to source control and publish.

3.5.4 Setup authentication details

This model class is used to setup authentication details for a specific party type. The class member is responsible for saving the party data as well as providing the system name of an engagement action type (and associated note text) which will then be created and added to the current engagement by the calling process.

The system name of the member **MUST** be the same as the system name of the party type.

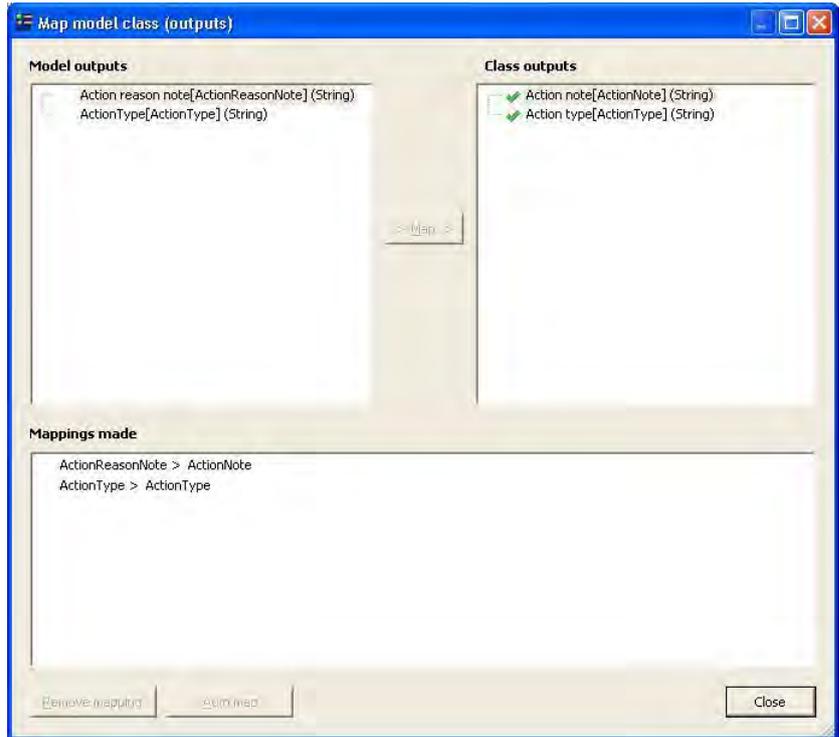
The **model class supports an 'OK', 'Cancel' and a 'Fail' outcome. In the take authentication transfer context, the 'Cancel, 'Fail' and the 'Not a member' outcomes will cause the process to terminate. It is imperative; therefore, that you supply a working member if you intend to use the Setup authentication details model class.**

The caller must supply the ID and the type of the party for whom the authentication details will be setup.

- 1 Create an overlay on the Setup authentication details model class.
- 2 Add a class member (system name **MUST** be the same as the system name of the party) referring to a business operation model that will request and then save the authentication details. Remember, you must add a working member if you intend to use the setup process



- 3 Map the class member inputs (in this case, the inputs to the model should be the Party ID and Party type of the party for whom the authentication details are being setup) by selecting **Map model class inputs...** from the class member context menu.
- 4 Create model class outcome mappings for each of the outcomes by selecting **Create model class outcome mapping...** from model class outcome maps context menu.
- 5 Map the member outputs for the OK outcome by selecting **Map model class outputs...** from the outcome map context menu (The member model should output the system name of a valid engagement action type and note text that will be stored in the engagement action).



- 6 Validate.
- 7 Add the overlay to source control and publish.

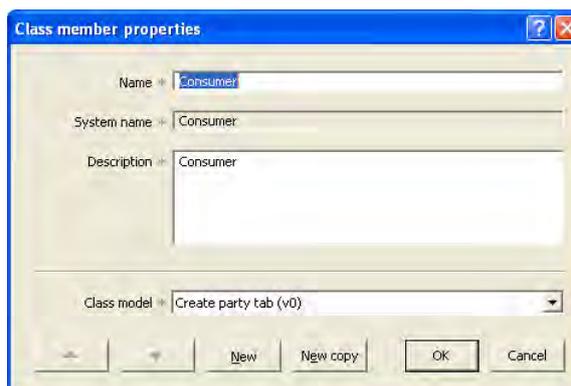
3.5.5 Add primary layer authority tab

This model class is used to add primary layer tabs for an authority. The member must have the same system name as the system name of the party type for the party granting the authority.

The model class supports an 'OK' and a 'Fail' outcome. In this context, the 'Fail' and the 'Not a member' outcomes will result in no tabs being added to the desktop.

In spite of the name, there is no specific reason why the member must add tabs at the primary layer (first visible row). Adding tabs at any layer will be subject to the maximum number of tabs per row limit (9) that applies to all models that manipulate the desktop.

- 1 Create an overlay on the **Add primary layer authority tab** model class.
- 2 Add a class member (system name MUST be the same as the system name of the party) referring to a business operation model that will add tabs at the appropriate layers to represent a party to whom an authority has been granted. Remember, you must add a working member if you intend to use the authorities process



- 3 Map the class member inputs (in this case, the inputs to the model should be limited to values than can be mapped from an Authority data object in addition to channel and language) by selecting **Map model class inputs...** from the class member context menu.
- 4 Create model class outcome mappings for each of the outcomes (there are no outputs to be mapped) by selecting **Create model class outcome mapping...** from model class outcome maps context menu.
- 5 Validate.
- 6 Add the overlay to source control and publish.

4 Add CBE views

By default, the Contact Centre uses Party views to populate the tertiary layer tabs within the **Engagement desktop** of the application. This behaviour can be modified in an implementation. If you intend to construct the desktop in another way, this section may be irrelevant.

This section lists views that may be added to a party definition when used in conjunction with other packages some of which will already be included in the *Contact centre workspace*. For some of the views it may be necessary to explicitly include other packages (depending on which packages are included by your Party package).

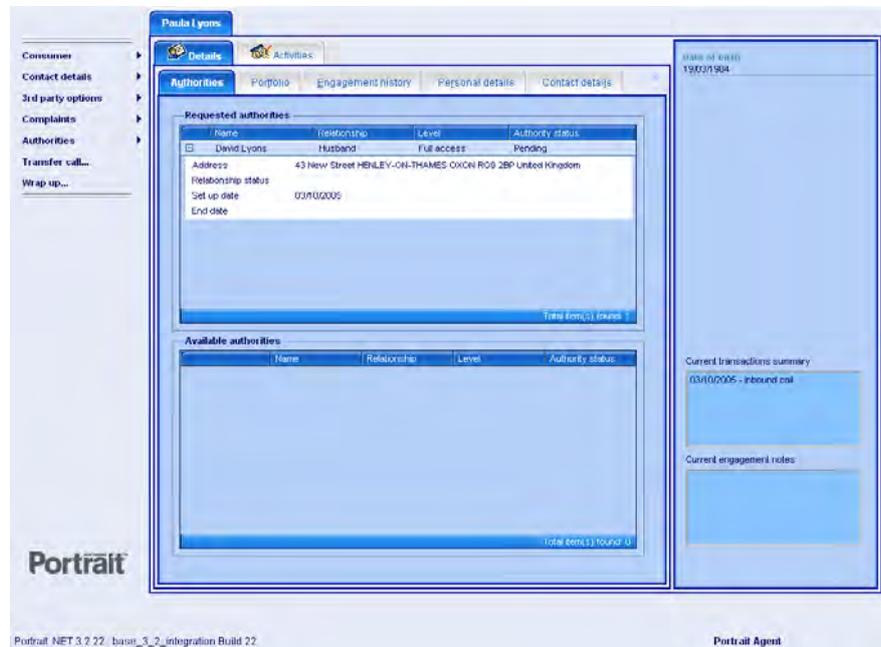
This section assumes that your Party definition already contains a **Business view** and any other generated view pages that display attributes of the party (e.g. **Personal details**).

4.1.1 Authorities

The Authorities view is used to display two grids. Each grid lists relationships between the current party (identified by the selected primary tab) and other parties.

An **Authority** is an *authorising* relationship between two parties. The top grid shows authorities that the current party has granted to other parties. The grid is used to modify or end authorities.

The bottom grid shows authorities that have been granted to the current party by others. This grid is used to execute functionality on behalf of those parties.



To add this view:

- 1 Create an overlay on the party views folder that applies to your party definition (e.g. Consumer). If no folder exists, then create the folder.
- 2 Open the *Base* party views folder.
- 3 Copy each of the following view definitions (if required) from the *Base* party view folder and paste it into your overlay.
 - a Authorities

- b Contact details
 - c Engagement history
 - d Portfolio
 - e Summary
 - f Workflow
- 4 Validate.
 - 5 Add the overlay to source control and publish.

5 Controlling party search

Functionality has been provided in Portrait Platform to enable projects to search for different party types. This functionality enables projects to configure what attributes to use as search criteria and display in subsequent result columns on the **Search for party** (custom interaction) and make use of the existing identify party configuration. The attributes selected must be defined as searchable.

The screenshot shows a dialog box titled "Attribute properties (Read only)". It contains the following fields and options:

- Name: Surname
- System name: Surname
- Description: The individual's surname (last name).
- Options:
 - Used in search: True
 - Attribute alias: (empty)
 - Is Unique: False
 - Data type: String
 - Data size: 254
- Buttons: New, New copy, Close

Data defined as searchable is stored additionally in specific Portrait database tables that have been designed for optimum efficiency. An attribute can subsequently be amended to be searchable, but records that were created prior to this change will not be found by a search on that value. If this is a customer requirement then a procedure could be developed to identify these records and populate the appropriate search tables in the Portrait database.

A party can be found by matches to data held in their party definition, any associated repeating attributes or contracts.

Assumption: Party definition inherits from Portrait Platform Base party.

5.1 Party attribute search criteria and results view

This view is configured specifically for a given party type to enable different criteria to be defined for different party types.

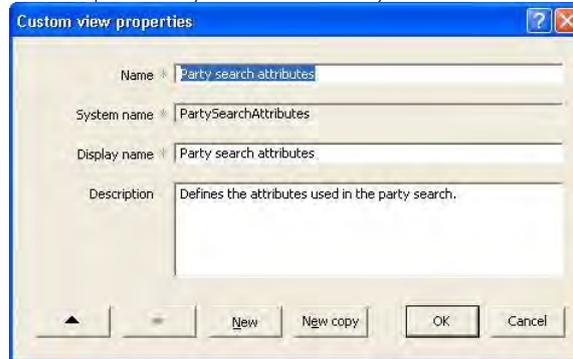
This configuration can be undertaken in the package where the party definition has been defined or a package that contains an overlay or override to an out of the box party definition e.g. **Consumer**.

5.1.1 Party search attributes view

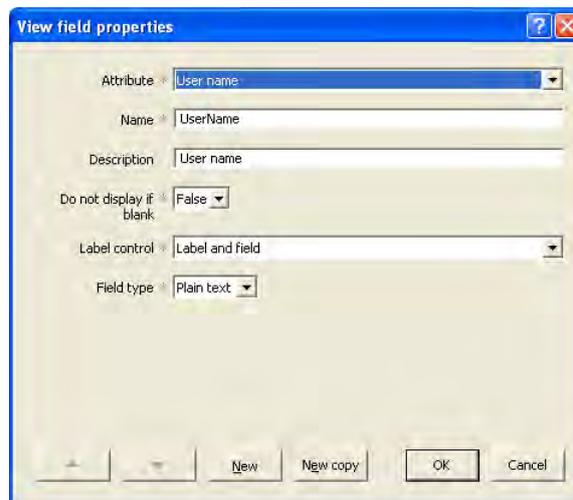
View functionality has been utilised to enable project implementations to define their own criteria to search for a party. Party view definitions can be configured for each party type.

- 1 Open package via Portrait Configuration Suite.

- 2 Expand **User interface**→**Views**→**Party view folders**. Choose appropriate party CBE or add by right clicking **Party view folders** and selecting **New party views folder...** then pick required CBE type.
- 3 Open appropriate party view folder and either select existing **Party search attributes** custom view or add via **New custom view...** (ensure system name equals PartySearchAttributes).



- 4 Expand and select **Party search attributes**→**View fields** right click and select **New view field...**
- 5 **New view field** dialog will appear. Select an attribute (ensure it has been defined as searchable) to be displayed as search criteria on the **Search for party** (custom interaction) and complete required fields. Repeat this step for all required attributes via the **New** or **New copy** functionality.



- 6 Validate.
- 7 Add to source control and publish.

5.1.2 Results view

To define the columns that you wish to display on the search results grid undertake the following steps:-

- 1 Open package via Portrait Configuration Suite.
- 2 Expand **Parties** and select the appropriate party definition. Open this item by double clicking or right clicking and selecting the **Open item** menu option.
- 3 Expand Attribute summaries and select Search summary view.
- 4 Right click and select **New view field...**

- 5 **Search summary view field properties** dialog will appear. Select an attribute that is required to be displayed in the search results view and complete the required fields. Repeat this step for all required attributes via the **New** or **New copy** functionality.



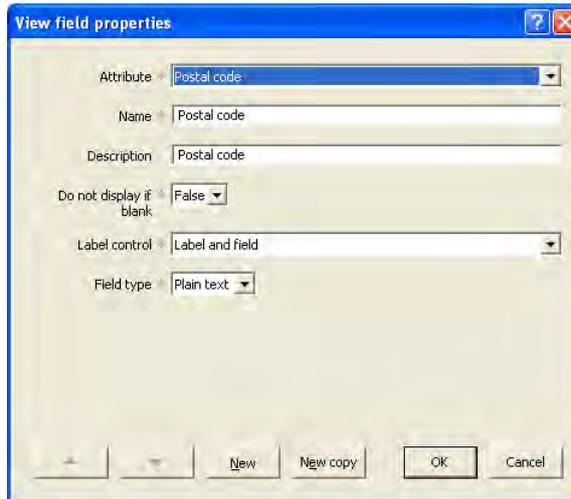
- 6 Validate.
- 7 Add to source control and publish.

5.2 Repeating attribute search criteria

This configuration can be undertaken in the package where the repeating attribute has been defined or a package that contains an overlay or override to an out of the box definition e.g. Address.

View functionality has been utilised to enable project implementations to define their own criteria to search for a party. View definitions can be defined for multiple repeating attributes. Portrait Platform configuration relies on the repeating attribute view folder being configured with the same system name as the required repeating attribute object definition.

- 1 Open package via Portrait Configuration Suite.
- 2 Expand User interface→Views→Repeating attribute view folders. Choose appropriate CBE or add by right clicking Repeating attribute view folders and selecting **New repeating attribute views folder...** then pick required CBE type.
- 3 Open appropriate view folder and either select existing **Party search attributes** custom view or add via **New custom view...** (ensure system name equals PartySearchAttributes).
- 4 Expand and select Party search attributes→View fields right click and select **New view field...**
- 5 **New view field** dialog will appear. Select an attribute (ensure it has been defined as searchable) to be displayed as search criteria on the **Search for party** (custom interaction) and complete required fields. Repeat this step for all required attributes via the **New** or **New copy** functionality.



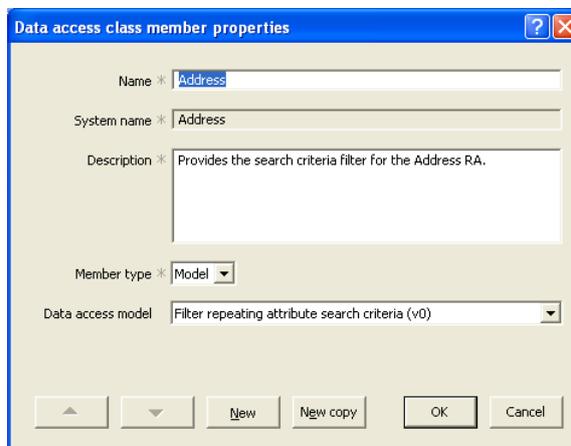
- 6 Validate.
- 7 Add to source control and publish.

5.2.1 Filter RA search criteria data access class

To plug in to the existing Portrait Platform configuration an appropriate class member will need to be added for each repeating attribute that has attributes to be used in a party search.

Assumption: Project implementation package includes (directly or indirectly) Parties (Supporting utilities) package.

- 1 Open package via Portrait Configuration Suite.
- 2 Expand **Supporting definitions**→**Models**→**Data access classes**. Select and open **Filter RA search criteria** data access class. Check out existing overlay or create a new overlay.
- 3 Expand and select **Class members** right click and select **New**→**Data access class member...**
- 4 **Data access class member properties** dialog will appear. Select **Filter repeating attribute search criteria** data access model. Complete required fields ensuring that the system name of the class member equals that of the repeating attribute.



- 5 Right click on the class member and select **Map data access class inputs...**. Add mappings.

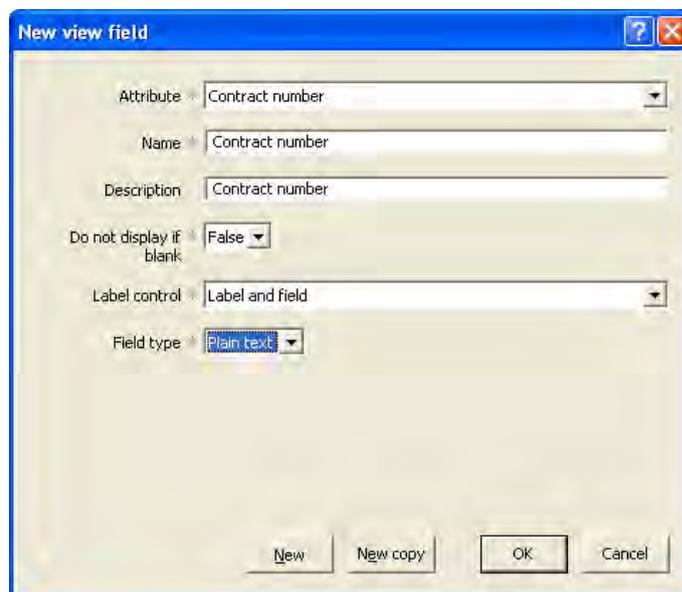
- 6 Right click on Data access class outcome maps and select **Create data access class outcome mapping...**. Add mapping. Repeat this step for each outcome.
- 7 Select OK outcome and right click then choose **Map data access class outputs...**. Add mapping.
- 8 In an appropriate package that contains the required availability criteria for the data access class i.e. party type. Select the class member, right click and choose **Set member availability...**. Create an appropriate circumstance for the requisite leaf party type(s).
- 9 Validate.
- 10 Add to source control and publish.

5.3 Contract attribute search criteria

This configuration can be undertaken in the package where the contract definition has been defined or a package that contains an overlay or override to an out of the box definition e.g. **Base**.

View functionality has been utilised to enable project implementations to define their own criteria to search for a party.

- 1 Open package via Portrait Configuration Suite.
- 2 Expand **User interface**→**Views**→**Contract view folders**. Choose appropriate contract CBE or add by right clicking **Contract view folders** and select **New contract views folder...** then pick required CBE type.
- 3 Open appropriate view folder and either select existing **Party search attributes** custom view or add via **New custom view...** (ensure system name equals PartySearchAttributes).
- 4 Expand and select **Party search attributes**→**View fields** right click and select **New view field...**
- 5 **New view field** dialog will appear. Select an attribute (ensure it has been defined as searchable) to be displayed as search criteria on the **Search for party** (custom interaction) and complete required fields. Repeat this step for all required attributes via the **New** or **New copy** functionality.



- 6 Validate.
- 7 Add to source control and publish.

5.3.1 Filter Contract search criteria data access class

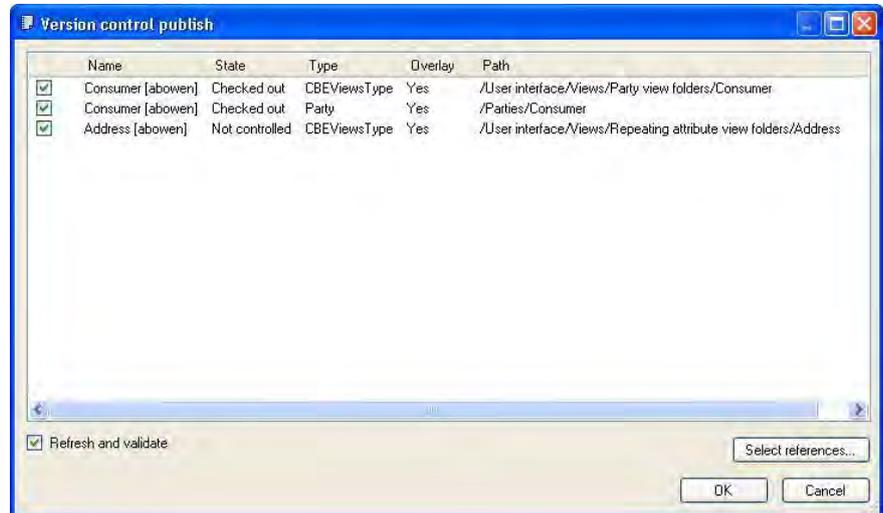
To plug in to the existing Portrait Platform configuration an appropriate class member will need to be added for the leaf contract business entity.

Assumption: Project implementation package includes (directly or indirectly) Parties (Supporting utilities) package.

- 1 Open package via Portrait Configuration Suite.
- 2 Expand **Supporting definitions→Models→Data access classes**. Select and open **Filter Contract search criteria** data access class. Check out existing overlay or create a new overlay.
- 3 Expand and select Class members right click select **New→Data access class member...**
- 4 **Data access class member properties** dialog will appear. Select **Filter Contract search criteria** data access model. Complete required fields ensuring that the system name of the class member equals that of the contract definition.
- 5 Right click on the class member and select **Map data access class inputs...** Add mappings.
- 6 Right click on Data access class outcome maps and select **Create data access class outcome mapping...** Add mapping. Repeat this step for each outcome.
- 7 Select OK outcome and right click then choose **Map data access class outputs...** Add mapping.
- 8 In an appropriate package that contains the required availability criteria for the data access class i.e. party type. Select the class member, right click and choose **Set member availability...** Create an appropriate circumstance for the leaf party type(s).
- 9 Validate.
- 10 Add to source control and publish.

6 Publish and Deploy

You should publish all the configuration changes that have resulted from the steps described in this document and then deploy the changes into a suitable database.



Before you can run the integrated functionality it will be necessary to create a suitable Portrait installation using the Portrait Build Kit. (See **Error! Reference source not found.**)

Appendix A Integration Checklist

This appendix should be printed so that the integration steps can be checked off. The checklist identifies the high level steps, each of which will contain several details steps.

You must complete the detail steps in the order indicated in the document. It is recommended that you complete the high level steps in the order indicated in this appendix. However, more experienced users will recognise that this isn't always necessary.

A.1 Mandatory steps

You must complete the following high level steps before you can use the packaged functionality.

Check the detailed steps carefully, they are not all mandatory.

Step	Description	Page	Done
1.	Add package	8	
2.	Set operation availability	9	
3.	Contact centre menu items	11	
4.	Ensure party type searchable	12	
5.	Add party views	20	
6.	Define party search criteria and results view	22	
7.	Publish and deploy	28	
8.	Test		

A.2 Optional steps

The following steps allow you to customise/extend your implementation further. If you undertake any of the steps you must complete the detailed steps as indicated in the document. **Do not partially complete an optional step.**

Step	Description	Page	Done
1.	Add class members	13	