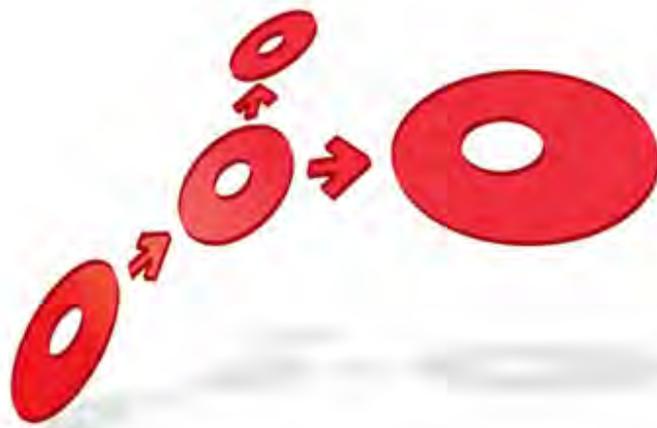


Portrait Foundation



Live Updates User Guide

Edition 6.0

11 January 2013



 **Pitney Bowes**
Software



Portrait Foundation Live Updates User Guide For Configurers and Deploymnt Administrators

©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

The aim of this document is to explain how the Live Updates functionality affects configuration and deployment procedures.

Intended audience

This document is aimed mainly at Configuration Suite users and System Administrators but also at anyone wanting an understanding of the Live Updates functionality.

Related documents

The Configuration Suite and Deployment tool on-line help.

Software release

Portrait Foundation 3.1 or later.

Contents

1	Introduction	6
2	Recap of Versioning	7
2.1	Versioned and non-versioned items	7
3	Environment types	8
3.1	Production	8
3.2	Non-production	8
4	Updates to Configuration Suite and other tools	9
4.1	Deployment wizard	9
4.2	The Deployer window	12
4.3	Model Diagnosis Tool	15
5	Configuration Guidelines	16
5.1	Configuration and release processes	16
5.2	Guidelines for maintaining non-configuration collateral	17
6	Releasing to Production	19
6.1	Steps to deploy and activate a new release	19
6.2	Cancelling a Deployment	21
7	Portrait System Administration	22
7.1	System downtime	22
7.2	Setting up servers in Portrait Management Console	22
7.3	Monitoring performance of Portrait Caches	23

1 Introduction

Prior to Portrait Live Updates the runtime system had to be made unavailable whilst new configuration data was deployed to the production database. This meant that all existing sessions had to be terminated. The configuration could then be deployed and the system restarted again.

The Portrait Live Updates functionality allows configuration changes to be deployed without impacting current sessions or without the need for downtime (except in rare circumstances).

Configuration changes can be rolled out whilst a system is live but without affecting running sessions. The latest configuration is made available to a user when a new session is started, that is, when a Portrait Foundation application is started. This ensures that an application remains consistent throughout its lifetime.

The functionality includes:

- Session orientated rollout of configuration updates
- A deployment wizard
- Detailed deployment history
- Delayed activation of deployments
- **Cache statistics available through 'PerfMon' counters**

This document describes the features listed above and how they affect Configuration Suite and Deployment users.

Some background information about versioning is provided for readers that are unfamiliar with these issues or are new to using Portrait Foundation.

The document also provides more detailed information for administrators of live Portrait Foundation systems, including guidelines on how updates to other collateral such as document templates, web pages and other supporting files can be rolled out effectively.

A summary of the sections is listed below:

- Recap on Versioning
- Naming conventions for non-configuration collateral
- Portrait Foundation runtime environments
- Updates to the Configuration Suite and supporting tools
- The development phase – Notes for Configuration Suite users
- Releasing to Production
- Notes for Portrait Foundation Administrators

It is important to note that the concept of 'Release numbers' introduced in 2.6 has been superseded by the 3.0 support for Live Updates.

2 Recap of Versioning

2.1 Versioned and non-versioned items

The high-level nodes in a Portrait Foundation workspace fall into two categories – versioned and non-versioned.

Non-versioned nodes typically describe the data structures used in the system. For example, parties, products, custom data objects, reference data.

Versioned nodes are usually those that have some kind of data interface (inputs, outcomes, and outputs) associated with them, and are used chiefly for defining business processes and logic. For example, Business Process Models, Scripts, Custom Interactions.

In the workspace there can be multiple versions in existence at any one time. This allows an item to be modified or evolved over time, but the dependants of that item can be upgraded independently.

For example, a Custom Interaction (CI) for capturing callback information could be used by two different business processes. If one of these processes needs an extra piece of information to be captured by the CI, for example the name of a particular agent to make the call, then a new version of the CI could be created, but only the business process affected needs to be modified. The other process would continue to use the original version of the CI.

During the development and maintenance of a Portrait Foundation system it would be inconvenient if the version number of an item is automatically incremented whenever an item is modified (checked-out of source control), because of the knock on effect on dependant items. Therefore, under the current support for Live Updates, a specific version can be modified and deployed repeatedly without having to change the version number.

With this new capability comes the additional requirement that the change should **[eventually] be compatible with the all the dependant's use of the item at the time of the next deployment** i.e. the changes made eventually make a matching set.

The point here is that changes to specific versions (e.g. an 'interface') can be made without changing the version. This technically makes the system undeployable until the corresponding changes are made to the users of the interface. Once all of the items have been aligned with the new interface the system can be deployed because integrity has been restored.

It is an error to change and attempt to deploy a version of an item if changes to that version are not compatible with the current state of other items within this deployment. Attempts are made to enforce this 'compatibility' by more rigorous validation at deployment time.

Each deployment will create a new instance of the item in the database. The unique key for this instance is a combination of the deployment number, version, and system name. The runtime (via its many caches for these items) will select the appropriate instance based on the deployment id assigned when the session was started, and the system name and version provided as in the past.

3 Environment types

The Portrait Foundation database installation prompts the user for the environment type for the database.

Portrait Foundation supports two environment types – Production, and non-Production although these may be extended in future releases.

These values are used by the deployment process to determine whether:

- 1 versionable items in the database can be modified
- 2 a release name can be associated with the deployment
- 3 to show warnings about the sensitivity of the data held in the database.

3.1 Production

A production environment is one which holds the critical data for an organization about its customers, transaction history, workflow, etc.

Because Portrait Foundation applications are chiefly driven by configuration data it is critical that it remains **compatible with both the business data (customer's details, product details, etc.)** and other related configuration data.

The deployment wizard will detect that the environment that is about to be deployed to is a production environment, and will display a warning message (see the deployment wizard section in the next chapter).

3.2 Non-production

A non-production environment is a development or test database where the validity of the data is not as critical.

4 Updates to Configuration Suite and other tools

4.1 Deployment wizard

The deployment wizard is triggered by selecting 'Deploy' from the deployable nodes tab in the deployment window of the Configuration Suite or Deployer tools.

The wizard validates that the target database is compatible with the loaded workspace and allows the user to confirm that they wish to continue deploying to that database.

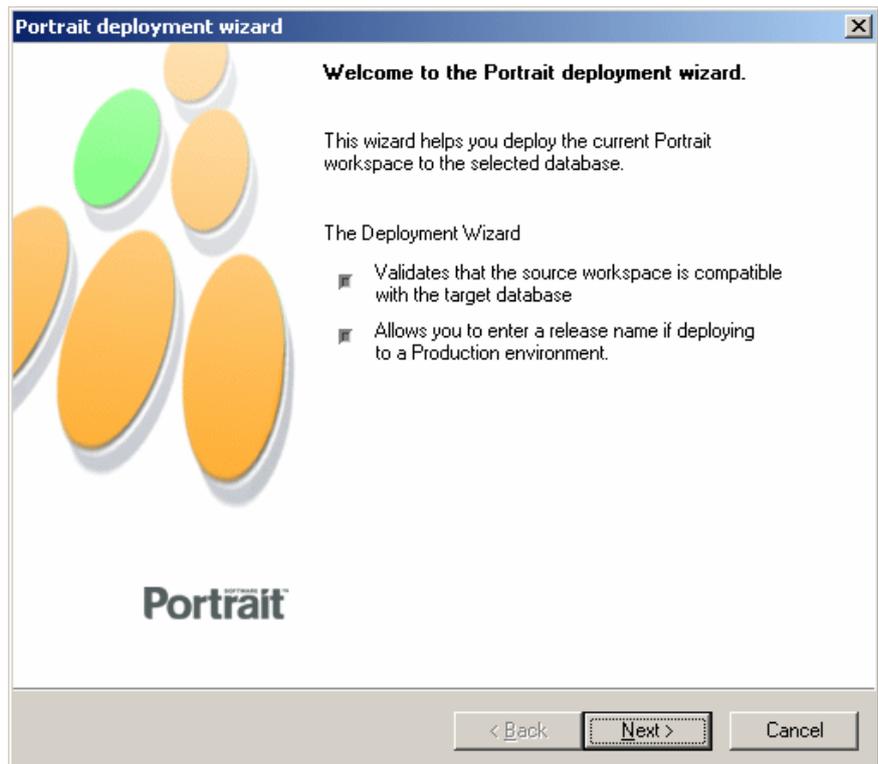
Checks are made to ensure that:

- 1 There are no Pending deployments that need to be activated before a new deployment can be performed.
- 2 That there are no other deployments taking place on the database.
- 3 The workspace id in the database and loaded workspace are identical.

4.1.1 Production Environment pages

Introduction Page

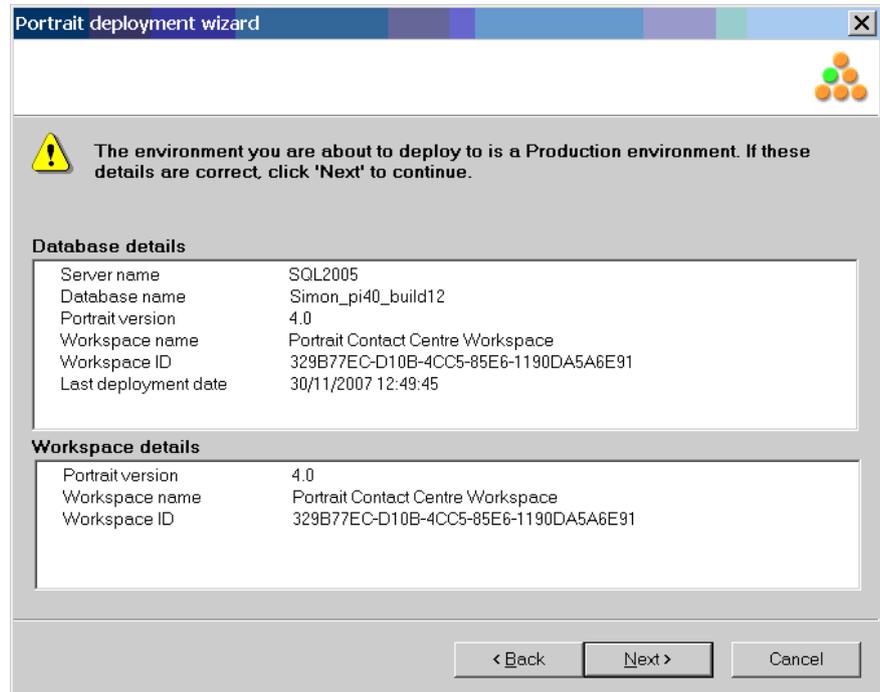
Figure 1 - Deployment Wizard - Introduction page



This is the welcome page.

Confirmation of target database

Figure 2 - Deployment Wizard - Confirmation of target database (Production)



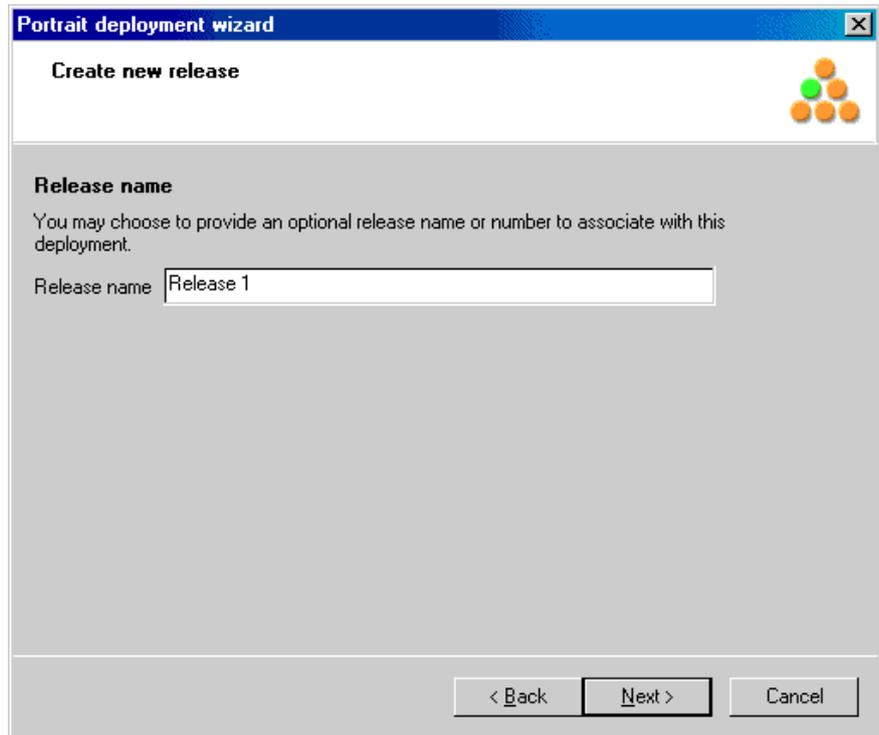
A warning icon is displayed to alert the user that they are about to deploy to a production environment. If there are any incompatibilities between the workspace and the target database the items are highlighted where applicable and an error message is displayed.

The following validation rules are applied in a production environment and if any errors are found a message will be displayed at the top of the dialog and the deployment will not be allowed to continue:

- The workspace id – the id of the workspace and the workspace that is deployed to the database must be equal. If there has been no workspace deployed to the database (new database) then any workspace id can be deployed.
- Portrait Software version number – The major and minor software release numbers must be identical.
- **The workspace must be set to show the 'Published' view because only published items can be deployed.**
- Last deployment status – the last deployment made must be activated before further deployments can take place.

Release name

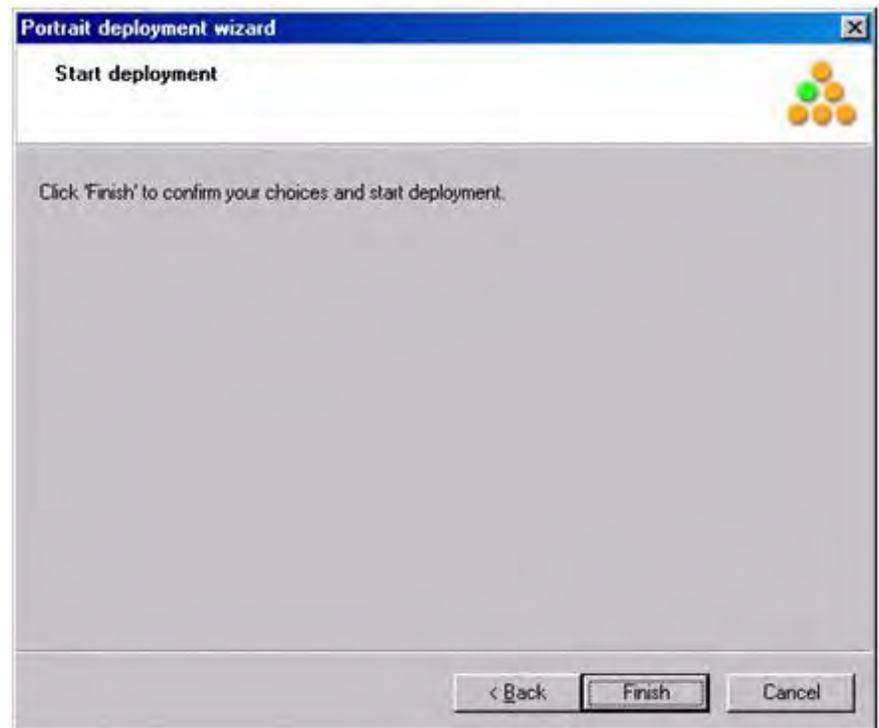
Figure 3 - Deployment Wizard - Release name page



This page allows a release name to be associated with the deployment.

Start deployment

Figure 4 - Deployment Wizard - Start deployment page



This is the last page and deployment starts when the finish button is clicked.

4.1.2 Non-production pages

In a non-production environment the order of the pages is as follows:

- 1 Introduction Page
- 2 Confirmation of target database – The validation rules are slightly different for non-production. The differences are :
 - Draft items can be deployed
- 3 Start deployment

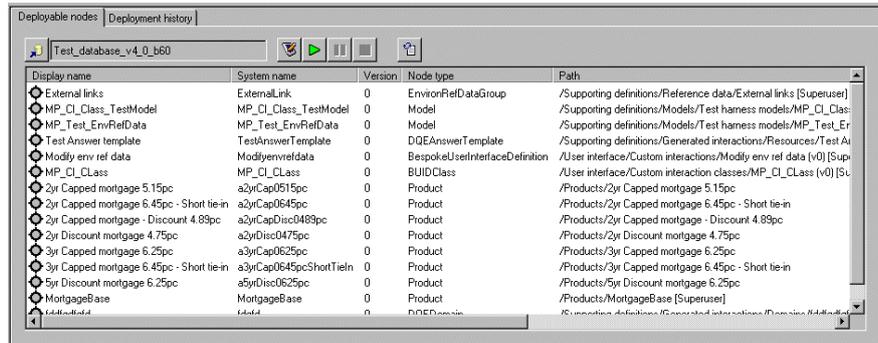
4.2 The Deployer window

The Deployer window is displayed in the Configuration Suite (by selecting View/Deployer) and from the Deployment tool. It has two tabs:

- 1 Deployable nodes
- 2 Deployment history

Deployable nodes tab

Figure 5 - Deployable nodes tab



The deployable nodes tab shows the information in a consistent format with the deployment history tab.

The options available from this tab are:



Select database

Availability Available when there is no deployment in progress.

Action Displays the database selection dialog.

Shortcut key ALT+S



Deployment options

Availability Available when there is no deployment in progress.

Action Displays the deployment options dialog (Full, necessary mode etc).

Shortcut key ALT+ G



Start deployment

Availability Available when there is no deployment in progress or if a deployment is currently paused.

Action Starts or restarts the deployment process.

Shortcut key ALT + D



Pause deployment

Availability Available when a deployment is in progress.
Action Suspends deployment until start or stop is selected.
Shortcut key ALT + P



Stop deployment

Availability Available when there is deployment in progress or the deployment is paused.
Action Stops the deployment - all database updates are rolled back / cancelled.
Shortcut Key ALT + T

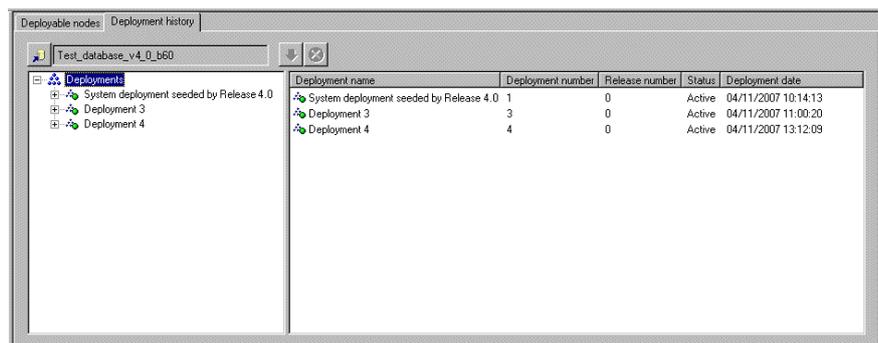


Full Refresh

Availability Available when there is no deployment in progress.
Action Refreshes the list of deployable nodes.
Shortcut key ALT + R

Deployment history tab

Figure 6 - Deployment history tab



The view is split into two main panes. The pane on the left (tree view) shows all deployments that have been made to the database. The name shown will be what the user entered as a release name if it is a production environment or Deployment N if a non production environment. Each deployment has two child nodes. The first is set of all nodes deployed during that deployment (Updated Nodes) and the second is the set of all deployed nodes that are in context for that deployment (Relevant Nodes). By selecting each of these folders the set of applicable nodes will be shown in the right hand list view pane.

The options available from this tab (depending on the item selected) are:



Select database

Availability Available when there is no deployment in progress.
Action Displays the database selection dialog.
Shortcut key ALT + S



Activate

Availability This function is available if the item selected has a status of Pending.
Action The deployment wizard performs the 'Activation' steps. See deployment wizard above.
Shortcut key ALT + A



Cancel

Availability	This function is available is the item selected has a status of Pending.
Action	The selected deployment is cancelled. All related configuration data is removed from the database.
Shortcut key	ALT + C

4.3 Model Diagnosis Tool

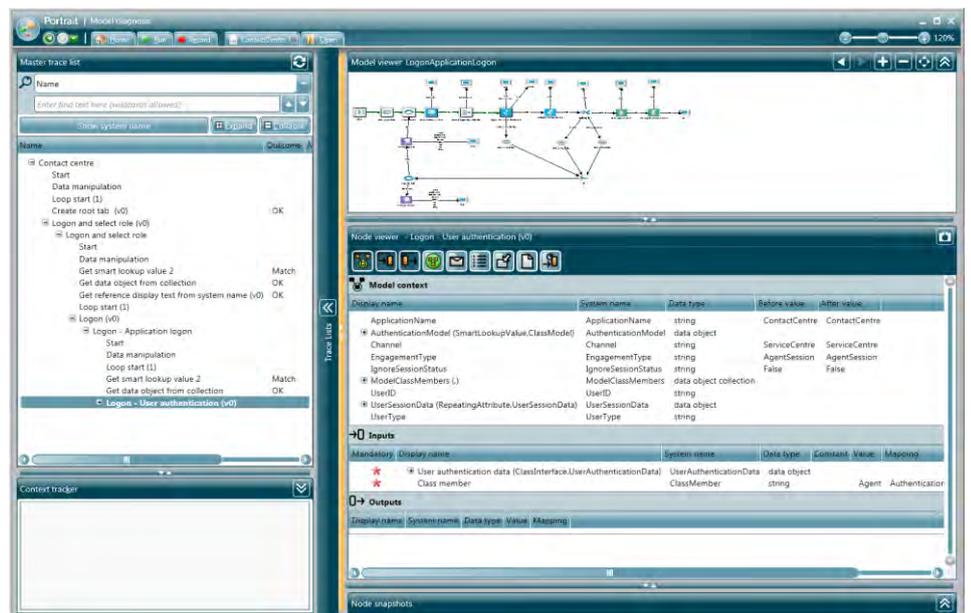
The data displayed by the model diagnosis tool includes details of:

- Model system name
- Model version
- Session id
- Activity token
- Deployment number

This information is available by selecting a 'model' node in the left-hand tree view. The information is displayed in the data tree view on the right-hand side.

As the mechanism by which deployment number is assigned is effectively bypassed when using the model diagnosis tool (the tool does not use a channel enabler), the tool allows the user to select which deployment number they wish the model to run with. The default is the deployment flagged as being the 'latest'.

Figure 7 - Model execution details in the Model Diagnosis Tool (MDT)



5 Configuration Guidelines

5.1 Configuration and release processes

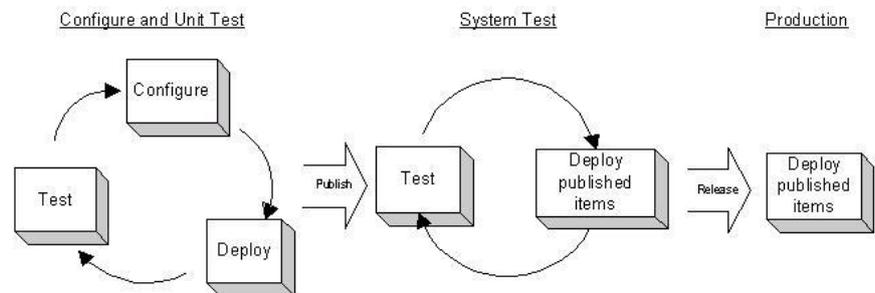
Each deployment of configuration to a production environment is considered to be a new definition of the system or a 'Release'. A release might include minor changes to reference data, new business processes or a completely new application definition.

Regardless of the size and impact of the changes made the recommended process remains the same: Some configuration changes are made to the master workspace by one or more configuration suite users, the changes are tested and published, system tested in a non-production environment and then finally released to production. The duration of each of these phases is usually determined by the quantity and complexity of the changes made.

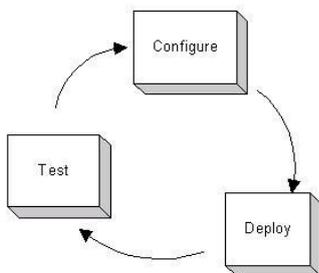
Each time a workspace is deployed the items deployed become available with the new deployment id as part of their unique key. This protects the running system from changes that can affect in-flight models or other instance data – any in flight model will continue to use the deployment id that it was originally started with. This even applied to very long running tasks such as workflow process models – these will use their original deployment number and activity token throughout their lifetime.

The process for making modifications to a workspace and then applying those changes to a production environment is described below. This section focuses on those aspects that concern configuration suite users. The stages involved are summarised in the diagram below:

Figure 8 - Steps involved in each environment



5.1.1 Configuration changes and unit testing



This stage is where the changes to configuration and supporting collateral (see section Guidelines for maintaining non-configuration collateral below).

The recommended development / configuration environment is to have a local unit testing database for each configuration suite user and the runtime components and services installed on the local machine. This way draft versions of configuration can be deployed and tested before they are published and made visible to everyone else in the team. Also, this isolates each user from being impacted by other user's changes.

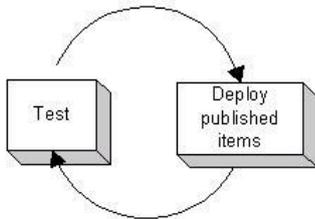
The alternative is to make configuration changes, check them in and publish them before deploying to a shared database and runtime environment.

After a successful deployment, the changes are activated immediately (this step includes updating any search and unique values if applicable). There is no need to restart services as the caches are automatically refreshed during activation.

Once the unit of work has been successfully unit tested the item(s) can be published and made available to everyone.

5.1.2 Publish

When item(s) are published they can be deployed and accessed by other members of the team. This could include system testers and other configurers. Further work may be carried out on the item(s) before the final version of the system is complete.



3 System Test

System testing, or acceptance / integration testing, should be performed on the published view of the configuration workspace.

System testing of completed areas of functionality can be performed whilst further development continues.

Once the final version of the system is published final acceptance testing can be performed before the configuration changes are applied to production and a new release is created.

5.1.4 Release

Once the configuration has been successfully system tested it can be released to production. Because development and production environments are usually separate from each other (physically on different networks and geographic locations) it is unlikely that the deployment to production would be performed on the same copy of the master workspace as that used in testing. That is, at some point a copy will need to be made.

5.1.5 Start next release

Configuration work on the next release of the system can be started at any time.

Deleting versioned items

When a versioned item that is obsolete is removed from configuration and has been deployed in a previous release it will not be removed from the database as previous versions of the system that might still be running may be making use of it.

(A separate task to purge the database of unused configuration must be performed – This functionality is not yet available but should be available in subsequent releases.)

5.2 Guidelines for maintaining non-configuration collateral

Non-configuration collateral includes DATS, Custom Nodes, other components that are packaged in DLLS/EXEs, Word or other document templates, Web pages (.asp, .aspx, .xslt, images etc).

In order to allow new versions of these items to be updated in a live environment without impacting the running system the way they are named and packaged is important. This document provides some guidelines on how best to do this.

In the current version of Live Updates a version of an item is no longer immutable. There may be several deployments of it that vary on their interface. The only guarantee provided by the deployment process is that the configuration for a given deployment is compatible. It does (and cannot) check that the new deployments are compatible with the non-configuration collateral. This introduces the need to guard against incompatibilities by naming and packaging.

5.2.1 Source Files

In order to allow the system to be updated without impacting the running servers the best approach is to always version files by including a version number in the name either by using a version number that relates to a versionable item in configuration or if the item is not versioned then to the release in which it was introduced. If a project wishes to allow the changing of an interface without changing a version then a naming strategy that includes version and release (which is effectively deployment) needs to be introduced.

For example,

If we had version 1 of a Custom Interaction (which is version in configuration) called Logon, we would call the physical file Logon_v1.aspx.

If we had a document template that contained quote information to be sent to the customer then it would be better to use the release number because document definitions are not versioned in the configuration suite. E.g. we would call the document used in release 1 : Quote_r1.doc.

5.2.2 DLLs, Exes and other binary files

For DLLs the name should probably relate to the release. Each DAT should have a new CLSID and Prog ID (named by configuration suite version number) and be packaged in a new DLL (named by release number).

As with source files; if a project wishes to allow the changing of an interface without changing a version then a naming strategy that includes version and release (which is effectively deployment) needs to be introduced.

For example,

If we had a DAT called CreateQuote, in release 1 it would be:

CLSID = {A guid}, ProgId= XXX.CreateQuote.1 and the DLL would be something like MyProjectCustomDATs_r1.dll.

In release 3 we might introduce a new version of the DAT, this would then be packaged as:

CLSID = {a new guid}, ProgId = XXX.CreateQuote.2 and the DLL would be called MyProjectCustomDATs_r3.dll

5.2.3 Summary

Always ensure you can copy the new or changed files onto a server whilst the server is running. Any new or updated files should not be used by older versions of the system.

6 Releasing to Production

At the end of a development phase or after any kind of configuration changes the workspace will be deployed to the production environment.

Before this is done a number of things need to be considered:

- 1 Have changes to search and/or unique settings on existing attributes of Parties, Products, Repeating Attributes, Engagement Actions or Significant Events been made? If so then it is likely that the system will need to be made unavailable (taken down) whilst the necessary database updates are performed.
- 2 **Has the 'Name view' (Used to generate the `known_as` column on `amc_pce_party_type_data` and `amc_pce_contract` tables) of any or party or contract been updated – if so this will require database maintenance to be performed and the system to be taken down.**
- 3 Are there any new supporting files that need to be applied to specific servers in the production environment? For example, new Web pages, XSLT, Microsoft Word Documents, DLLs etc. These files will need to be installed on the appropriate servers before any new configuration that might be dependent on them is activated.

It is recommended that the steps are carried out in the following order when creating a new release. Not all of the steps will need to be performed depending on the nature of the changes made.

6.1 Steps to deploy and activate a new release

6.1.1 Step 1 - Install supporting files

Copy or install all new supporting files such as web pages, xslt, scripts, dlls, word or other document definitions to the appropriate servers (Web server, print server, CRM server etc.). If the guidelines outlined in section 5.2 have been followed then it should be possible to copy files to the server without impacting the running system. There should be no dependencies on any of the data held by these implementation files from the configuration in the running system so they will not be used until the new configuration is activated. If the steps in section 5.2 have not been followed and existing files have been modified then it may not be safe to overwrite the existing file whilst the system is running. This all depends on the nature of the change and the type of file being copied.

6.1.2 Step 2 - Deploy the updated configuration

- 1 Open the workspace in the configuration suite or deployment tool
- 2 **Select the 'Published' view setting so that only published items are shown.** The view is set automatically to Published in the Deployment Tool.
- 3 If using the configuration suite, display the Deployer view.
- 4 Make sure that the Deployer settings are set to deploy only items that have been modified or are new – **'Necessary' deployment mode.**
- 5 Click on the start deploy button.
- 6 Follow the steps in the deployment wizard, enter the name or number of the release.
- 7 If deployment completes successfully the changes will be committed but the deployment status will be set to Pending. If no database maintenance needs to be performed then the deployment can be activated immediately.

Otherwise it should be left as Pending and the steps to take the system down should be followed before activating the deployment.

If database maintenance **does not** need to be performed the release can be Activated immediately and users will gradually move on to the latest release as new sessions are started.

If database maintenance **does** need to be performed then the remaining steps need to be followed.

6.1.3 Step 3 - Downtime to allow database maintenance to be performed

If the system needs to be taken down before the new deployment can be activated you will be notified at the end of deployment. If the system is still live (running) then it will need to be taken down.

This can be done in a number of ways and is dependant on the environment and implementation.

When taking the system down a number of things need to be considered:

If the system is purely a contact centre environment then the best time to do this would be out of business hours unless 24x7 operation is mandatory in which case there needs to be some fall back system in place to cover for system unavailability.

If there are self service channels, access to new sessions needs to be prevented and users informed leading up to the planned downtime. Any sessions that are active should be allowed to complete if possible.

In order to control the planned unavailability of the system extra functionality could be built into the configured applications, for example controlling the operation availability through some criteria that can be managed through an administrator tool. This could force users to log off at specified times.

All servers that access configuration need to be taken down. This includes Web Servers, CRM servers, Print servers etc.

The system needs to end up in a state where all Portrait Foundation services are stopped on all servers and cannot be accessed until the maintenance is completed.

6.1.4 Step 4 - Activate the deployment and perform any database maintenance.

Once the system is down then the deployment can be activated. This can be **done through either the configuration suite's deployment view or using** the deployment tool.

From the Deployment History tab select the deployment (release) that has a status of Pending and click on the activate button. You will be reminded that database maintenance is about to be performed – **select 'yes' to proceed and the** necessary updates will be performed and the status will be set to active.

The system can now be restarted.

6.1.5 Step 5 - Restart the system

The services on each of the servers can now be restarted. All new sessions will begin using the latest deployment.

6.2 Cancelling a Deployment

If a deployment has been made to a production environment and it still has a status of pending then it can be cancelled and all configuration updates associated with that deployment are deleted from the database.

This can be done by selecting release with a status of Pending in the Deployment **History tab in the Configuration Suite's Deployer view or from the Deployment Tool** and clicking on the Cancel button.

Once a deployment has been activated it cannot be cancelled. The only course of action that is available is to effectively create a new release which contains the fix following the same process as outlined above.

7 Portrait System Administration

This section is aimed at administrators responsible for managing a live Portrait Foundation system.

7.1 System downtime

Current functionality does not quite cater for live updates of all configuration data. There are some circumstances where system down time is still required in order for updates to be carried out.

7.1.1 Search and Unique changes

When the search and / or unique settings of an attribute are changed. This applies to attributes of Parties, Products/Contracts, Repeating attributes, Engagement actions and Significant events. This is because the search and unique tables need to be re-built in the database. This can be a lengthy operation depending on the number of records in the main data table.

Also, it does not make sense to have two versions of the system running in parallel where one is saving searchable attributes and the other one is not – this could cause data to be lost, or in the case of unique values – invalid duplicates to be created. It is essential, therefore, that all sessions are using the latest deployment when the system is restarted.

7.1.2 Name view

System downtime is required when the Name view definition is changed on a Party or Contract. This view is used to construct the KNOWN_AS column on the party data and contract data tables. Obviously if the definition changes then all of these values need to be re-created. The value is generated whenever a party or contract is created or modified.

Again, it will cause inconsistencies in the data if two versions of the system are writing different formats of the KNOWN_AS field when updates are made to a party or contract.

7.1.3 Changing the database

Portrait Services still need to be restarted if the database is changed by modifying the settings in the AMC.udl file.

7.2 Setting up servers in Portrait Management Console

There are some changes in the way that servers are set up using the management console. In the past it was an optional task to add each server into the management console. This was only done when settings for a particular machine needed to be modified.

It is essential that all servers are configured in order for system notifications to work correctly.

The reasons why this is essential and the impact of not doing so is described below.

7.2.1 Active Deployment Number

Portrait System Configuration stores a value for the latest active deployment number. The value is stored in :

Category : DEPLOYMENT

Sub-category : Number

To avoid the possibility of notifications of new deployments not being delivered correctly the latest deployment number is retrieved from this table (via the System Configuration component) whenever a new session is started.

7.3 Monitoring performance of Portrait Caches

There are a number of 'PerfMon' counters available which can be used to monitor the usage of various caches within the system. The caches that provide such counters are :

- Smart Lookup
- Configurable Business Entity definitions (CBE)
- Reference data
- Operations
- Custom interactions
- Post operation navigation definitions
- Input / Output definitions
- Mapping definitions
- Script definitions
- Custom Control definitions
- DAT Definitions

The values that are logged for each cache are :

Cache Size Limit (# items)	The limit set by in the Portrait Management Console, or zero if it is disabled.
Current Cache Size (# items)	The current number of items in the cache (zero if disabled)
Cache Hits / sec	The number of cache hits per second recorded by each cache.
Cache Misses / sec	The number of cache misses per second recorded by each cache.