



Multi-tenancy Feature Overview

Edition 3.0

03 July 2014





Portrait Foundation Multi-tenancy Feature Overview

©2014
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.](#)

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 introduces the Multi-tenancy feature in Portrait Foundation. The first part of this document highlights the changes to Portrait Foundation. The later sections go on to explain the operation of more than one Portrait system on a single server.

Intended audience

This document is intended for all users of Portrait Foundation, who intend to have more than one installation of Portrait on a single server. It is also useful background information for those solution providers who are developing integrations or extensions to Portrait Foundation, even if they are not developing solutions intended for use on a single server with more than one installation of Portrait Foundation.

Related documents

Installation Guide

Software release

Portrait Foundation 5.0 or later.

Contents

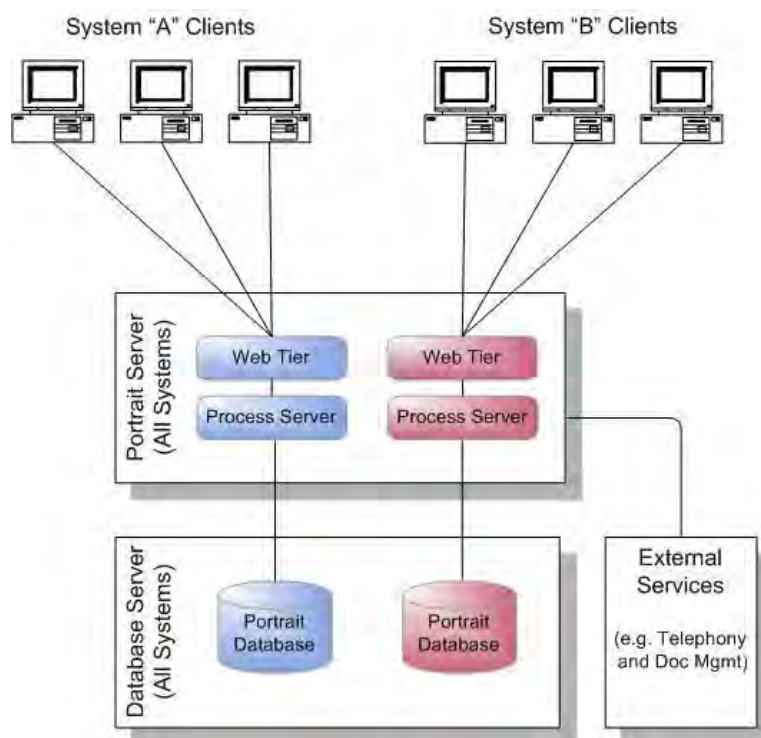
1	Overview of Multi-tenancy	6
1.1	Main features	6
2	Concepts of Multi-tenancy	8
2.1	Identification of a Portrait Foundation system	8
2.2	Independent systems	9
3	A multi-tenancy installation	10
3.1	Installation	10
3.2	System management	11
3.3	System Configuration	12
3.4	Operation and administration	12
3.5	System performance	14
3.6	Integrations and extensions	15
4	Reference	17
4.1	Components	17
4.2	Services	17
4.3	Tools	18

1 Overview of Multi-tenancy

A multi-tenancy installation is designed to reduce the capital investment required for multiple installations of Portrait Foundation. Several installations can be hosted on a common server but seem discrete and independent **from the users'** point-of-view. The data is isolated between each installation to ensure that there is no crossover of any process or information.

A multi-tenancy environment provides for the coexistence of multiple instances of the same version of Portrait Foundation on a single physical server. The installation and management of each system is independent.

Figure 1 - Multi-tenancy architecture



The introduction of Multi-tenancy has brought some minor setup changes that are important for all solution providers to know.

1.1 Main features

Portrait Foundation with Multi-tenancy has the following features.

- All of the individual Portrait Foundation systems must share the same version of Portrait Foundation components.
- All of the individual Portrait Foundation systems share the same version of operating system and Microsoft components.
- There are no Configuration Suite changes in order to accommodate Multi-tenancy. All of the individual Portrait Foundation systems are configured, deployed and have external resources as separate entities.
- The Portrait Foundation systems on a server behave and coexist in exactly the same style as if the systems each had their own server.
- Each Portrait Foundation system acts as an independent entity and has no knowledge of the other Portrait Foundation systems. There is no cross checking of runtime configuration settings.

- Each individual system creation and administration is independent.
- The Portrait Foundation installation process is aware of the existence, on a server, of multiple Portrait Foundation systems.
- Although multiple Portrait Foundation installations share the same executables they do not share data; each installation has its own services and no shared runtime data.
- Each Portrait Foundation system on a server is a discrete system that has the ability to have its own security attributes.

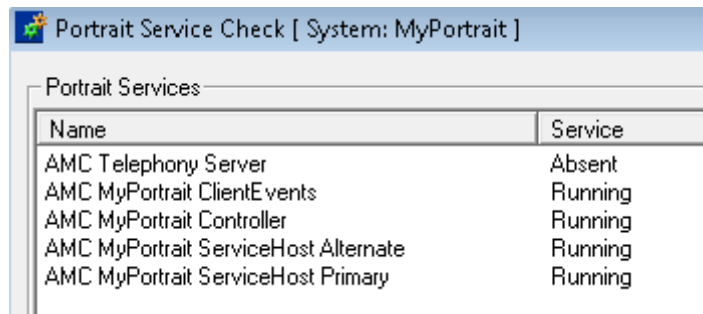
2 Concepts of Multi-tenancy

Multi-tenancy is achieved using an extension to Portrait Foundation’s flexible Service Host architecture. The Service Host architecture allows a configuration XML document to describe the components that are functioning within a given Windows service. This XML document lists Portrait Foundation components and links each of them to specific Portrait Foundation installation on the server. This elegantly isolates the Portrait Foundation components so that they are able to function effectively on a single physical server.

2.1 Identification of a Portrait Foundation system

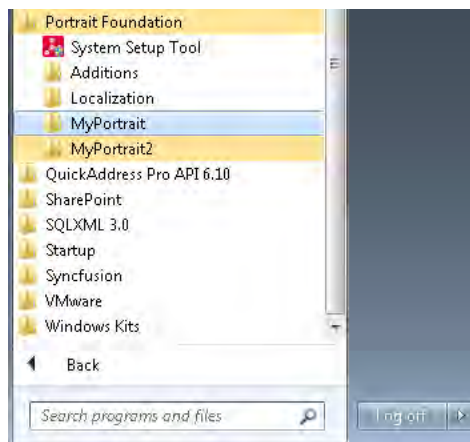
When performing an administrative action on a Portrait Foundation installation, it is necessary to identify the installation. As a specific example, if services are to be started or stopped for a named Portrait Foundation installation then the 'Service Check' tool will only show services that are for that installation. The title of the dialog box will indicate the named installation that is operated on at the time.

Figure 2 - Service check is working with the coromell system



Operational tools invoked from the taskbar **Start** menu, created as part of the install, will set the appropriate context. To make it easier for the user to distinguish the system context that they are in, an extra level of menu is created in the Windows Program group. These new shortcuts invoke the application with a command line switch indicating the Portrait Foundation system or context.

Figure 3 - Two Portrait systems



Irrespective of the number of Portrait Foundation systems installed on any one server, each system will have its own name. That is to say that if there is only one installation of a Portrait Foundation system on a server it will still be contained within a named installation.

2.2 Independent systems

The main Portrait Foundation executables are shared between all Portrait Foundation systems, but to ensure that there is no crossover of any process or information each system has its own:

- **Set of services that represent the system's runtime; for example Service Host and Web Channel Service.**
- Installation selected options.
- Each system has its own deployable configuration and operational data that can be hosted on the same device.
- Operational settings captured via the Portrait Management Console; these are held in the database and registry.
- Web Sites or virtual directories.
- Caches of state; for example model/session, configuration and smart lookups/ reference data.



As each Portrait Foundation installation shares the same Portrait Foundation code there cannot be any variation in version, service pack, update, or hot-fix. Any solutions that integrate or extend Portrait Foundation may need to be at the exact same version.

3 A multi-tenancy installation

3.1 Installation

The initial installation of a Multi-tenancy Portrait Foundation system starts with a normal Portrait Foundation installation. A detailed explanation of how to install Portrait Foundation can be found in the *Installation Guide*.



The maximum recommended number of multi-tenant systems on one server is 4.

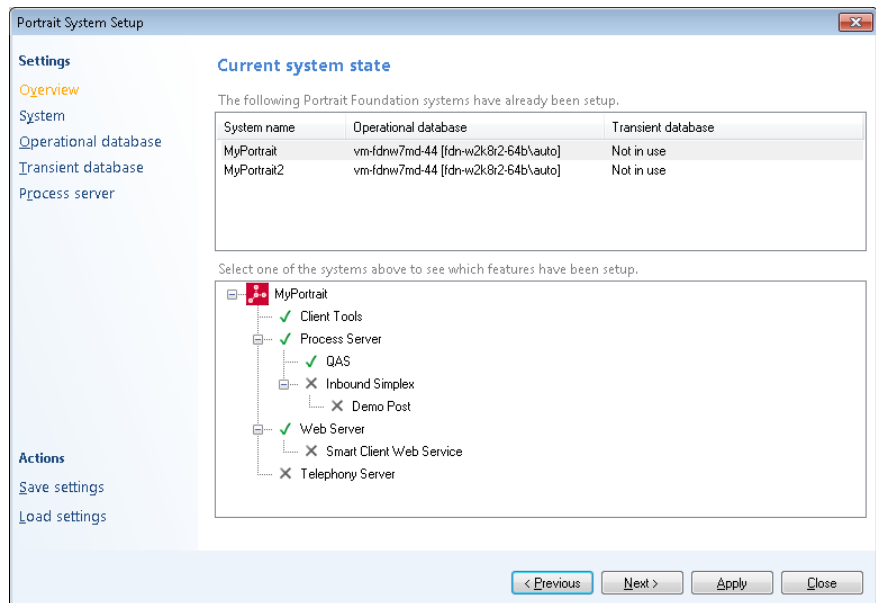
This section highlights how the changes to Portrait Foundation have changed the installation process and how additional installations of Portrait Foundation on a single server will differ.

3.1.1 Installation components

Each System on a device shares a common set of Portrait Foundation components. The Portrait Foundation Core Software install is responsible for installing and maintaining these common executables.

It is possible to add or remove components using the Foundation Core Software install at anytime. But once any changes have occurred each System will need to be updated using the System Setup Tool.

Figure 4 - System Setup Tool



3.1.2 Systems on a server

The System Setup Tool is the only mechanism provided to restructure a server in terms of adding, removing or updating Portrait Foundation systems.

It is possible to add or update systems in any order, as there is no significance of any particular system. In addition, removal of each system can be done in any order without affecting the capability of the server. However, it is recommended that all systems on the server be taken out of operation during a maintenance upgrade to ensure that all systems are the same version of service pack, update and hot-fix as described in *Independent systems* on page 9

Addition of any solutions that integrate with or extend Portrait Foundation can be done in any order once their named system has been installed. That is to say, all named systems can be installed then integrations added or a named system can

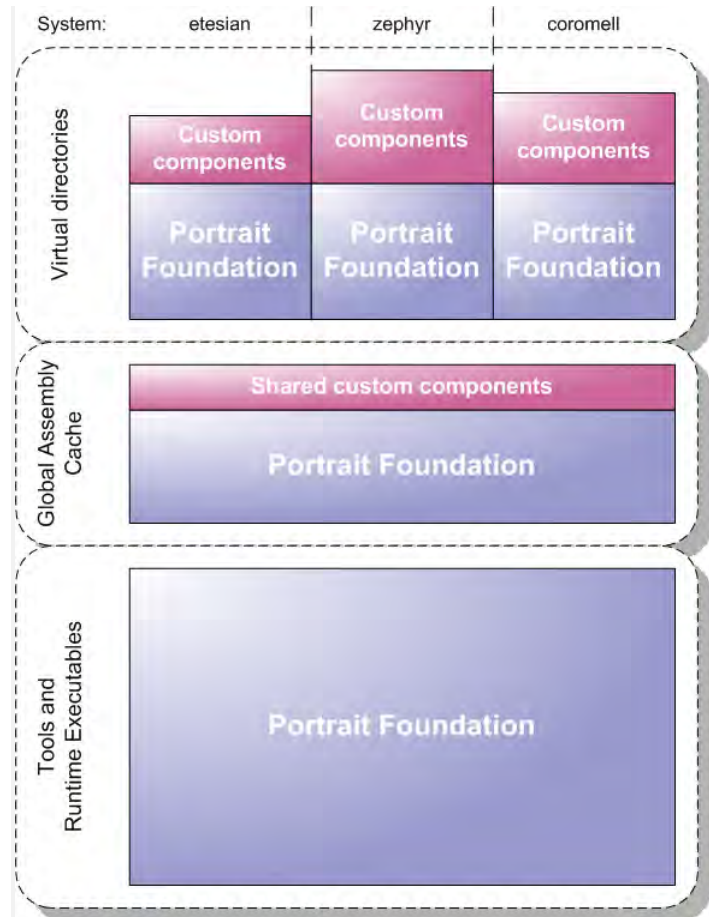
be installed followed by the related integration followed by another named system.

3.2 System management

There is little change in the approach to system management in the multi-tenancy environment.

A reference of the system components detailing their action in a multi-tenancy system is in the *Reference* section starting on page 17.

Figure 5 - Portrait shared components on disk



All Portrait Foundation systems will share the same version of operating system and Microsoft components, but all resources used by a Portrait Foundation system will be unique to their named installation.

Detail setup changes to Portrait Foundation are covered in *Integrations and extensions* on page 15.

3.2.1 Installed versions

As each Portrait Foundation installation shares the same Portrait Foundation executables there cannot be any variation in version, service pack, update, or hot-fix. Any solutions that integrates or extends Portrait Foundation may need to be at the exact same version.

3.2.2 Telephony

It is possible to have multiple Portrait Foundation systems use a single instance of the Portrait Telephony Server (PTS). The PTS must reside on its own server and

the configuration of the systems that share a PTS must use unique device numbers (DN's).

NB: It is not possible to have multiple instances of the PTS on any given server.

3.3 System Configuration

The number, name and function of most Portrait Foundation components on a system are dictated by the Portrait Service Host configuration XML document. Editing the configuration XML document is the only way to split-up, consolidate or rearrange the distribution of Portrait Foundation components.

The capability to divide a Portrait Foundation system between several physical servers remains fully supported with the introduction of Multi-tenancy. In addition, each system can have a different setup.

3.4 Operation and administration

Generally, there is little impact on the operation and administration of a Portrait Foundation installation with the introduction of Multi-tenancy, apart from the need to isolate Web Service worker processes for each system when running a Portrait Foundation 4.0 (or later) implementation; single system installs and earlier versions of Portrait Foundation are unaffected.

Each Portrait Foundation system acts as an independent entity and has no knowledge of the other Portrait Foundation systems. There is no cross checking of runtime configuration settings.

For example, if you want to check that there are no conflicts in the Client Event settings for two systems, you need to do a manual check—list each system's Client Event settings and make a manual comparison.

Administration of each individual system is independent.

3.4.1 Web applications in multi-tenancy systems

If you have a Portrait Foundation Version 4.0 (or later) implementation that runs Portrait web applications in a multi-tenancy system, you must make sure that each system has its own isolated IIS worker process. To achieve this, install all systems and then configure the web service:

IIS 7 & 8 - Allocate the services to different Application Pools

- 1 To create an application pool, create a new folder in C:\InetPub\WWWRoot and then open up Internet Information Server Manager from Administrative Tools.
- 2 Right-click the Application Pools node in the left pane and select New Pool. Enter a name for the pool, such as MyPortraitPool.
- 3 Right-click the newly created pool and select Properties from the dropdown to bring up the Configuration dialog box, where you can set the different parameters of the health of the pool.
- 4 Finally, add your application to the pool; right-click the virtual directory or folder under the website you have and select Properties. Click Create to make it a Web application if not already done. (In case the folder is an application, you will see the Remove button and not Create).
- 5 Set the Application Pool to MyPortraitPool in the same dialog box to add the current application to the pool created. This application will inherit the health parameters you have set.

3.4.2 Portrait Foundation tools

There is no change to the functionality or operation of Portrait Foundation tools to support Multi-tenancy. However, the named context in which the tool operates separates the use of each tool for a given Portrait Foundation system. To set this Portrait Foundation tools now support a command switch, `/System:YourSystemName`. A summary of Portrait Foundation tool operation is in *Tools* on page 18.

Some tools, like Configuration Suite, Deployer, Repository Manager and Data Mart Creation, have the capability of operating on any Portrait Foundation database or repository that they can access. This capability could cause some confusion if the operation of these tools is misunderstood.

When Configuration Suite is used to deploy to a different database than originally selected for the system in the context, the selection is saved and used by other services of that system. Obviously, this can have undesirable effects on the applications within the Portrait Foundation system.

Similarly, Repository Manager saves information about which repository is used by all configuration tools for the given Portrait Foundation system.

In addition, error logging for all tools is reported with the context of the Portrait Foundation system with which they are invoked. If you change the selection of repository or database this could potentially cause confusion of reported error messages.

It is, therefore, recommended that all tools be used within the named Portrait Foundation system that they are invoked to avoid any potential confusion.

3.4.3 Logging and diagnostics

Portrait Foundation logging operates on a specific Portrait Foundation system. Generally, the log viewer will capture data from the Portrait Foundation system within its own named context.

Eventlog

The Windows Eventlog destination has the name of the Portrait Foundation system in the **Source** column. This allows the identification and filtering of messages from different systems.

Figure 6 - Windows Event log shows the Portrait system

Tree	Source	Category	Event	User
Computer Management (Local)	zephyr	Portrait	44	N/A
System Trunk	zephyr	Portrait	45	N/A
Event Viewer	zephyr	Portrait	1010	N/A
Application	zephyr	Portrait	87	N/A
Security	zephyr	Portrait	16389	N/A
System	zephyr	Portrait	5	N/A
System Information	zephyr	Portrait	16389	N/A
Performance Logs and Alerts	zephyr	Portrait	4	N/A
Shared Folders	coromelly	Portrait	5	N/A
Device Manager	zephyr	Portrait	4	N/A
Local Users and Groups	zephyr	Portrait	16389	N/A
Storage	zephyr	Portrait	16389	N/A
Disk Management	coromell	Portrait	44	N/A
Disk Defragmenter	zephyr	Portrait	16389	N/A
Logical Drives	coromell	Portrait	1010	N/A
Removable Storage	coromell	Portrait	87	N/A
Services and Applications	coromell	Portrait	45	N/A

Other logging destination for example those that write disk files or database tables will create resources on a per Portrait Foundation system basis.

Perfmon

The viewing of Windows Perfmon counters will be unaffected by Multi-tenancy. The Perfmon application will report the counters against the instances of Portrait Foundation named services that they created. This will allow counters to be associated with a particular Portrait Foundation system.

Userdump

Diagnostic memory dumps via Microsoft's Userdump are an operating system feature and will be placed in a common location for all Portrait Foundation components on a physical server. Perfmon / task manager data will be required to determine to which system the process belongs. Portrait's MiniDump memory dumps will be produced in a location specific to a Portrait Foundation system and will be self describing.

Portrait Model Diagnosis Tool

The Model Diagnosis Tool will only diagnose models in the named system where it is invoked.

Service Host configuration

Each Service Hosted COM CLSIDS is unique. Capturing the Service Host configuration XML will aid problem resolution when these IDs are known.

3.4.4 System isolation

In the unlikely event that a named Portrait Foundation service or application fails, it will not influence the performance of any other Portrait Foundation system installed on the server.

3.4.5 Process resilience

Multi-tenancy systems that use Portrait Foundation 4.0 (or later) can each be independently set up to use different levels of process resilience.

3.5 System performance

Multi-tenancy will not have a significant impact on the performance of a suitably specified server.

Memory demands of multiple Portrait Foundation systems on a server grow in a linear fashion with the number of Portrait Foundation systems installed. Adding a second identical system would double the Portrait Foundation specific memory requirements over a single installation. This is because there are no shared data, configuration or caches between Portrait Foundation systems.

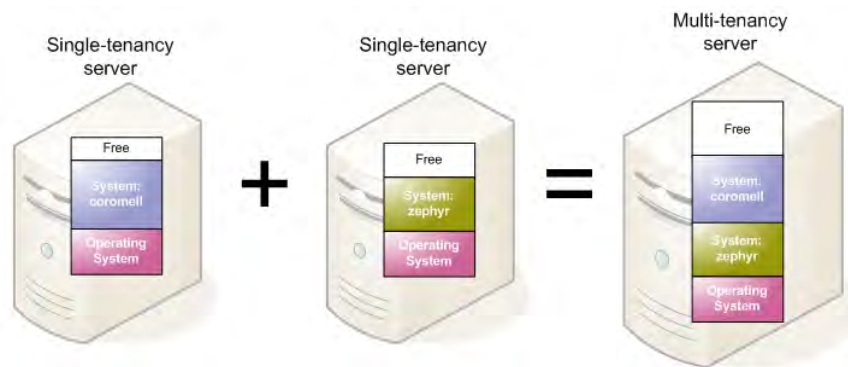
3.5.1 Planning memory requirements

Under normal circumstances, development of a new Portrait Foundation application would be done on a development server that has only one Portrait Foundation system installed.

Once development of a Portrait Foundation system on a single-tenancy installation is complete, it is important to understand the memory footprint of the solution. This information can then be used to plan the memory requirements that that solution will require on a multi-tenancy installation.

The figure below shows two development systems combined on a single production server. Understanding the memory footprint of the Portrait Foundation system on each development server allows sufficient capacity to be planned for the production server.

Figure 7 - Comparison of memory requirements for Multi-tenancy



3.6 Integrations and extensions

Multi-tenancy introduces considerations for those who develop solutions using Portrait Foundation.

3.6.1 System selection

When it is not possible to specify a system name in the command line for an integration or extension then the system name can be set by the API command `::SetSystemName()`. The system name is determined in the following order:

- Command line
- Environment variable `PortraitSystemName`
- Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\PST\Portrait` value `DefaultSystem`

3.6.2 Names and resources

Care is needed to ensure that names used for resources are unique within a server. Examples include.

- Operating system IPC objects (Mutexes/memory mapped files)
- Project specific COM ProgIds/CLSIDs
- Project specific .NET objects

Services shared by multiple systems will need to operate in a common Portrait Foundation system context. This is necessary because a shared service cannot be configured as part of any specific system on the server.

Any code that relies on Portrait Foundation registry locations or disk file locations will need to refer to a registry location or disk file location for a named Portrait Foundation system.

3.6.3 Registry keys

The Portrait Foundation registry keys have been restructured to identify specific systems. To access the registry keys and values for a system it is necessary to know which system the application is operating. The `GetRegistryKey()` PortraitLib API can be called from C++ or .NET and will return the registry key appropriate for your current system context.

3.6.4 Changes to integrations and extensions

Every integration or extension to Portrait Foundation will need to make the following changes; even if there will not be any plans to running multiple Portrait Foundation systems on a single server.

- Installation must specify a name for each Portrait Foundation system.

- Check any code that makes specific reference to
 - Windows Registry entries
 - file system components
- Any inter-process communication objects need to be reviewed.

Integrations and extensions to Portrait Foundation that will operate on a physical server that has more than one Portrait Foundation system will need to make these additional changes, to those mentioned above.

- Ensure that there is sufficient space on the server to run more than one Portrait Foundation system without severe negative impact to performance.
- Check that there are no conflicting runtime configuration settings between systems on a server.
- Each integration or extension to a Portrait Foundation installation will need its services to be hosted by Service Host or capable of having multiple copies of the same services running within a single server.

All integrations and extensions to Portrait Foundation must be fully tested before deployment to a production environment.

4 Reference

The following sections give specific details of how Multi-tenancy impacts Portrait Foundation components, services and tools.

4.1 Components

Component	Action
.NET versioning	Each solution that integrates or extends Portrait Foundation will need a .config file.
Channel adapters	Use web.config to determine system ownership. This name will then be used to identify the objects that are hosted in the Web Channel.
Logging	Logging is on a per system basis so the log viewer will look at messages for a particular system. Messages written to the eventlog identify the system that produced them under the 'Source' heading.
Object Broker	The Object Broker works within the context of a particular system. The objects that it are configured to be brokered will be the decorated names.
Process Server components	Each system will have its own named Service Host. A service component will determine which system it belongs to via command line arguments, <code>/System:YourSystemName</code> . Each service's components will be addressable by COM/DCOM as a unique ID that is 'decorated' with the systems name. For example. <code>AIT.AMC.Messaging.DCOMTransport</code> will become: <code>AIT.AMC.Messaging.DCOMTransport.SystemA</code>
Web Channel	The Web Channel, through the object broker wrapper, will connect to the appropriate Process Server components by using the decorated IDs. The Web Channel component will determine which system it belongs to via command line arguments. The Web Channel service is a Service Host component so uses Service Host's facilities to publish the components it hosts and to determine its system ownership.

4.2 Services

Service	Action
Automatic Tasks Engine	This has moved to a Service Host component.
Collector Service	This handles all Perfmon data for all systems.
Eastman Accessor Service	This has moved to a Service Host component.

Service	Action
Eastman DMS Inbound Adaptor Service	This has moved to a Service Host component.
Exchange DMS Inbound Adaptor Service	This has moved to a Service Host component.
Hint Server	This has moved to a Service Host component.
Log File Writer	This has moved to a Service Host component.
Log Message Concentrator	This has moved to a Service Host component.
Message Queue Listener	This has moved to a Service Host component.
QAS Service	This has moved to a Service Host component.
Service Host	Service Hosts have multiple instances each named after the system that they service. There will typically be one per system unless components are broken out into multiple services.
SysConfig Auxiliary service	This has moved to a Service Host component.
Telephony Server	Each system that uses telephony is required to have its own non-shared telephony server.
Web Channel Service	This has moved to a Service Host component.

4.3 Tools

Tool	Action
Batch Load Framework	Although a command line input will invoke in a named system, this application can operate on any Portrait database that it can access.
Configuration Suite	Although a command line input will invoke in a named system, this application can deploy to any Portrait Foundation database that it can access.
Data Mart Creation tool	Although a command line input will invoke in a named system, this application can operate on any Portrait Foundation database that it can access.
Log Viewer	This application has a command line input indicating in which system it will operate.
Model Diagnosis Tool	This application has a command line input indicating in which system it will operate.
Model and Node profiler	Although a command line input will invoke in a named system, this application can operate on any Portrait Foundation database that it can access.
Object Broker Microsoft Management Console snap-in	This application has a command line input indicating in which system it will operate.
Portrait Management Console Microsoft Management Console snap-in	This application has a command line input indicating in which system it will operate.
Profiling tools	This application has a command line input indicating in which system it will operate.

Tool	Action
Repository Manager	Although a command line input will invoke in a named system, this application can operate on any Portrait Foundation database that it can access.
Service Password reset	This application has a command line input indicating in which system it will operate.
Service Check	This application has a command line input indicating in which system it will operate.
Service Generation Wizard	This application has a command line input indicating in which system it will operate.
Version display tools	Although a command line input will invoke in a named system, this application can operate on any Portrait Foundation database that it can access.