Oracle® DIVArchive

WS API User Guide

Release 7.2

Legacy Document

August 2015

Oracle DIVArchive WS API User Guide, Release 7.2

Legacy Document

Primary Author: Lou Bonaventura

Table of Contents

Tables Index

Figures Index

What’s New in this Version

Section	Change	New / Update / Correction
Throughout	Rebranded to Oracle and published as Legacy Documentation.	Update

1 Introduction

1.1 Document Conventions

The following conventions are used with respect to text:

Normal	Standard Text.
Italic	Used to emphasize a term or variable.
Bold	Used to emphasize critical information.
6.1	Refers to a section or sub-section in the document.
`Courier New`	Used for system screen output and system commands.

The following conventions are used with respect to file paths or variables:

- `DIVAS_HOME`: The Root Path on the file system where DIVArchive WS is installed (*Default:* `C:\DIVAS`).
- `CARBON_HOME`: The path to the DIVArchive WS Application Server - **no changes should be made in this folder** (*Default:* `DIVAS_HOME\DIVAS\application-server`)!
- `JAVA_HOME`: The absolute directory path to the Java Runtime Environment (*JRE*) bin directory on the host.

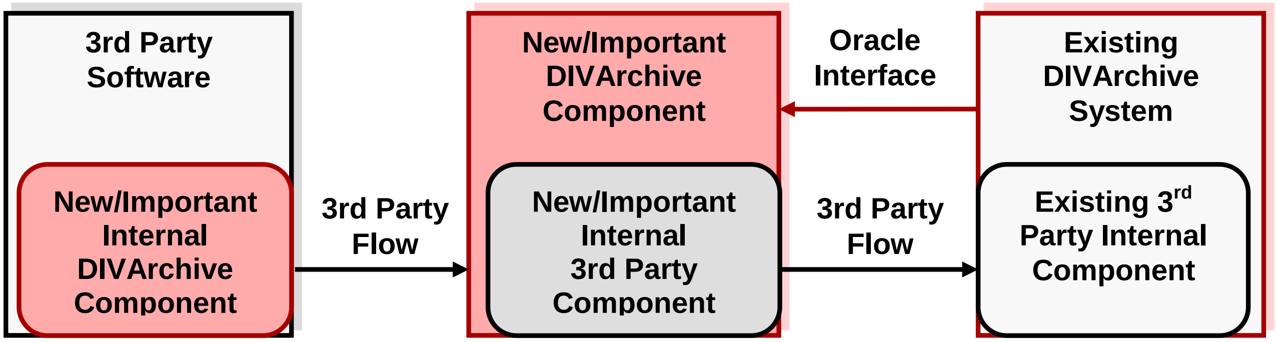
The following conventions are used with respect to figures and drawings:

Red outlined boxes pointing to specific areas in a figure indicate procedural steps, or point out specific parameters being discussed in the section text.

Red outlined boxes that surround specific areas in a figure indicate specific areas of the figure being discussed in the section text.

- For Oracle software components:
  - Oracle Red is used for borders and lines.
- For 3rd-party software:
  - Black and White are used for borders and lines.
- New/Important components:
  - Existing components have no shading.
  - New/Important components have light shading.

Figure 1: Non-Screenshot Drawing Displaying Drawing Styles


3rd Party Software
New/Important Internal DIVArchive Component
3rd Party Flow
New/Important DIVArchive Component
New/Important Internal 3rd Party Component
Oracle Interface
3rd Party Flow
Existing DIVArchive System
Existing 3rd Party Internal Component

1.2 Definitions, Acronyms and Special Terms

Table 1: Definitions, Acronyms, and Special Terms

Term	Definition
API	Application Programming Interface
Bundle	A group of software components (*DIVArchive WS 6.5, DIVArchive WS 7.0, etc.*).
Component	A single, integral element of the software bundle.
CRUD	Create, Read, Update and Delete (*CRUD*) are the four basic functions of persistent storage.
CygWin	A collection of tools providing a Linux look and feel environment for Windows.
DIVAS	A component of the Oracle DIVArchive Suite that includes the DIVApplication Server and the DIVArchive WS API Bundles.
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JVM	Java Virtual Machine: A piece of software that is implemented on non-virtual hardware and on standard operating systems.
REST	Representational State Transfer: Style of software architecture for distributed hypermedia systems such as the World Wide Web.
SOA	Service-Oriented Architecture: Flexible set of design principles used during the phases of systems development and integration in computing. A system based on an SOA will package functionality as a suite of interoperable services that can be used within multiple separate systems from several business domains.

Term	Definition
SOAP	Simple Object Access Protocol: A protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on XML for its message format, and usually relies on other application layer protocols, most notably RPC and HTTP, for message negotiation and transmission.
SOAPUI	A tool used for testing the Web Services.
WS	Web Services
WSDL	Web Services Description Language
XML	**Extensible Markup Language:** a text-based database used to allow for the easy interchange of documents and data on the World Wide Web or between software components (***.xml***).
.7Z	A compression protocol that is used to compress a single file, or multiple files, for easy transfer as a single archived file.

1.3 DIVArchive WS API Overview

The DIVArchive Web Services (*WS*) API is based on modern development technologies and methodologies providing flexible, scalable, and powerful interactions with the Oracle DIVArchive system using the WS standards.

The DIVArchive WS API is a combination of two architectures:

- Simple Object Access Protocol (*SOAP*): An extensible protocol for exchanging structured messages using the Web Services Description Language (*WSDL*).
- Representative State Transfer (*REST*): Takes a different approach than SOAP and follows closely to the Create, Read, Update, and Delete (*CRUD*) ideology rather than SOAP's extensibility.

DIVArchive has employed methods to ensure the full functionality expected using the SOAP methodology and the RESTful syntax. The DIVArchive WS API provides full CRUD functionality in addition to more advanced features available from both SOAP and RESTful implementations.

The DIVArchive WS API is written using the WSDL XML format. All of the definitions are contained in the **`.wsdl`** file and the schemas are in the **`.xsd`** files:

- **`DIVArchiveWS.xsd`**
- **`DIVArchiveWS_exception.xsd`**
- **`DIVArchiveWS_model.xsd`**
- **`DIVArchiveWS_response.xsd`**
- **`DIVArchiveWS_SOAP.wsdl`**

The DIVArchive WS consists of two parts:

- The DIVArchive Application Server Platform version 1.1.
- The DIVArchive Web Services Application.

And is compatible with the following bindings and formats:

- SOAP
  - SOAP 1.1
  - WSDL 1.1
  - Bindings:
    - HTTP, SOAP 1.1, (*no https*)
- REST
  - No WSDL required.
  - Bindings:
    - HTTP (*default*)

- WS-I
  - SOAP 1.1

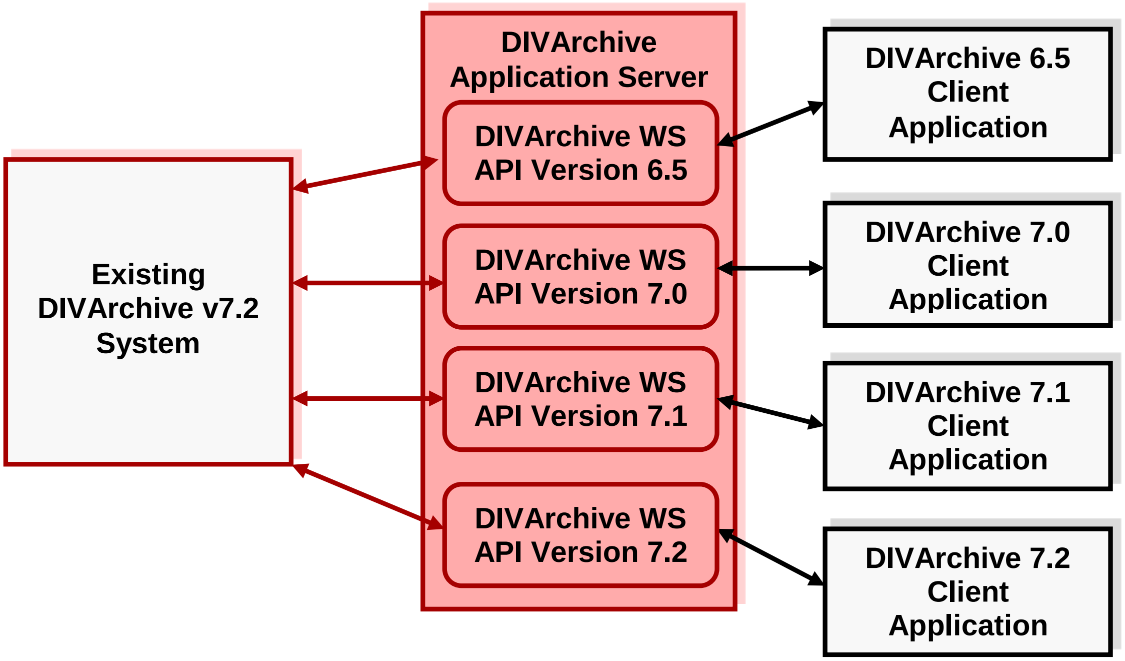
Note: After installation, the Schemas are located in:
`DIVAS_HOME\components\DIVArchiveWS.n.n\references`

Where `n` is the DIVArchiveWS version. Should the Schema be required before DIVAS installation, please contact the Oracle Support Department.

For more details on DIVArchive WS bindings and formats refer to the DIVArchive 7.2 WS API Reference Manual.

The figure below demonstrates how the DIVArchive WS Bundles integrates with DIVArchive version 6.5, 7.0, 7.1, and 7.2. The DIVArchive Application Server, DIVArchive WS 6.5, DIVArchive WS 7.0, DIVArchive WS 7.1, and DIVArchive WS 7.2 are delivered as **DIVAS 2.2**.

Figure 2: DIVAS 2.2 Integration with DIVArchive 6.5, 7.0, 7.1, 7.2


Existing DIVArchive v7.2 System
DIVArchive Application Server
DIVArchive WS API Version 6.5
DIVArchive WS API Version 7.0
DIVArchive WS API Version 7.1
DIVArchive WS API Version 7.2
DIVArchive 6.5 Client Application
DIVArchive 7.0 Client Application
DIVArchive 7.1 Client Application
DIVArchive 7.2 Client Application

1.4 DIVArchive Version Compatibility

A DIVArchive WS client is compatible with the same version of the DIVArchive WS API and the DIVArchive WS API is compatible with all current and future versions of DIVArchive. A DIVArchive WS client with version 6.5.x.x.x would be compatible with DIVArchive WS API version 6.5.x.x.x which would be compatible with any DIVArchive version 6.5, 7.0 and later.

Any new features added to DIVArchive after the version of the WS API in use will not be available; the client system must be upgraded to the latest version to utilize all features. Multiple DIVArchive WS installations may be installed at once and so each client system should connect to the server with which it is compatible.

Table 2: DIVArchive WS Compatibility

Delivery Package Version	DIVArchive WS Version	Compatible DIVArchive Version	Backward and Forward Compatibility
2.0.x	1.0.x	6.5.x	Backward compatibility is not supported. Forward compatibility: compatible with version 6.5 and higher.
2.1.x	2.0.x	7.0.x	Backward compatibility is not supported. Forward compatibility: DIVArchive 7.0.x has been deprecated, therefore forward compatibility is not supported.
2.2	2.1	7.1.x and higher	Backward compatibility is not supported. Forward compatibility: compatible with DIVArchive version 7.1.x and higher.

2 DIVArchive WS Installation

2.1 Prerequisites

The DIVArchive WS API has several requirements that must be met to allow interfacing with the DIVArchive System via Web Services:

- The standard DIVArchive prerequisites.
  - DIVArchive WS can be installed on any machine whether DIVArchive exists on the machine or not, however the standard DIVArchive prerequisite requirements must be met.
  - Refer to the DIVArchive 7.2 Configuration Guide (*Appendix D1.1*) for standard prerequisites.
- Java Virtual Machine (*JVM*) version 1.6.0_21-b07.
  - **`JAVA_HOME`** must be defined before the installation.
- **`7Z`** and **`CygWin`** must be installed (to support the log collection process)
  - Both programs are readily available on the internet.
    - http://www.7-zip.org/
    - http://www.cygwin.com

2.2 Installation Process

The installation of the DIVArchive WS API can be performed in two ways:

- Using the zip file.
- Using the exe file.

Using the exe file is similar to the standard DIVArchive 7.2 installation. DIVArchive WS can be installed on any machine regardless of whether DIVArchive exists on the machine or not. Executing the **`DIVArchiveWS.exe`** application will begin the installation process and allow the installer to select the components to be installed. In the event that the zip file method is being used, please unpack the file to the root folder and go to 0 for further instructions.

Multiple DIVApplication Servers may be run at one time throughout the system. However they must be on separate machines as only one (*1*) DIVApplication Server can run on a single machine at any time.

Each DIVApplication Server is capable of running one, or several, bundles (*of different types*). The currently available bundles are:

- DIVArchive WS API 6.5
- DIVArchive WS API 7.0
- DIVArchive WS API 7.1
- DIVArchive WS API 7.2

Two similar bundles cannot be run on a single DIVApplication Server.

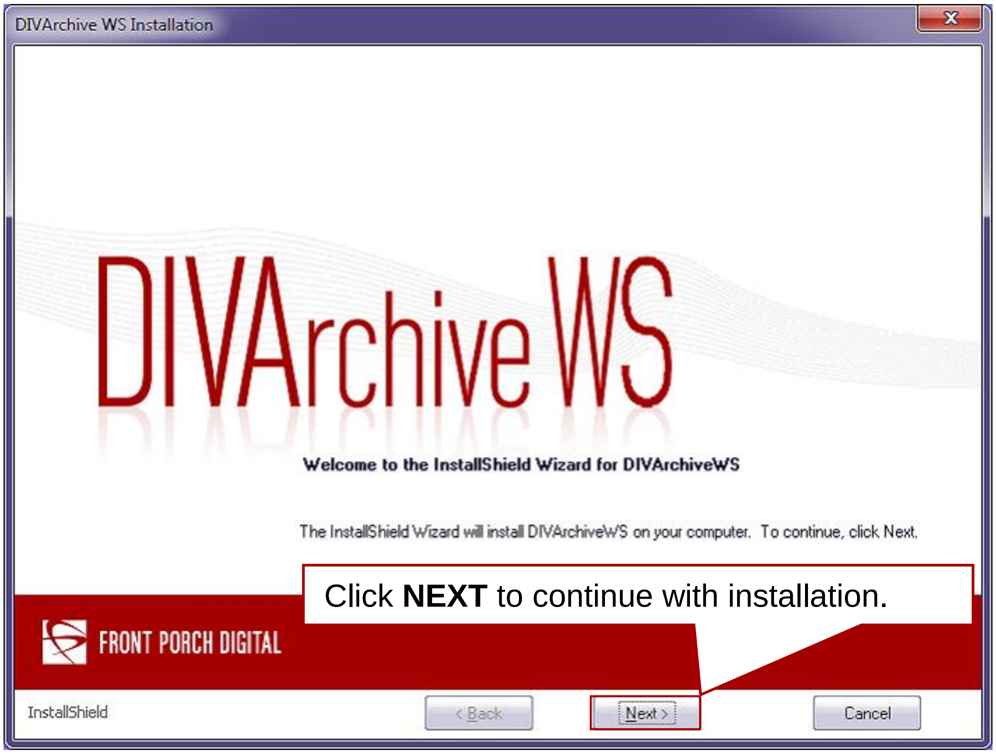
Executing the **`DIVArchiveWS.exe`** application will begin the installation process and allow the installer to select the components to be installed.

1. Start the Windows System Command Window (***`START`**, **`RUN`**, **`CMD`***)
   - The **`CMD`** window must be started using an administrator account.
2. Run the installation executable file (*depending upon your operating system, Administrator privileges may be required*).
   - **`DIVArchiveWS.exe`**
   - The full path must be used: **`[Full_Path_to_Files]\DIVArchiveWS.exe`**.

Note: Alternatively, the `DIVArchvieWS.exe` installation may be started by double-clicking the file in Windows Explorer rather than using the command line as explained in steps 1 and 2 above.

3. When the installation process begins, click the **NEXT** button at the bottom of the screen (*see the figure below*). Throughout the installation process the installer may use the **BACK** button to return to the previous screen.

Figure 3: Welcome Screen for DIVArchive WS Installation



Welcome to the InstallShield Wizard for DIVArchiveWS

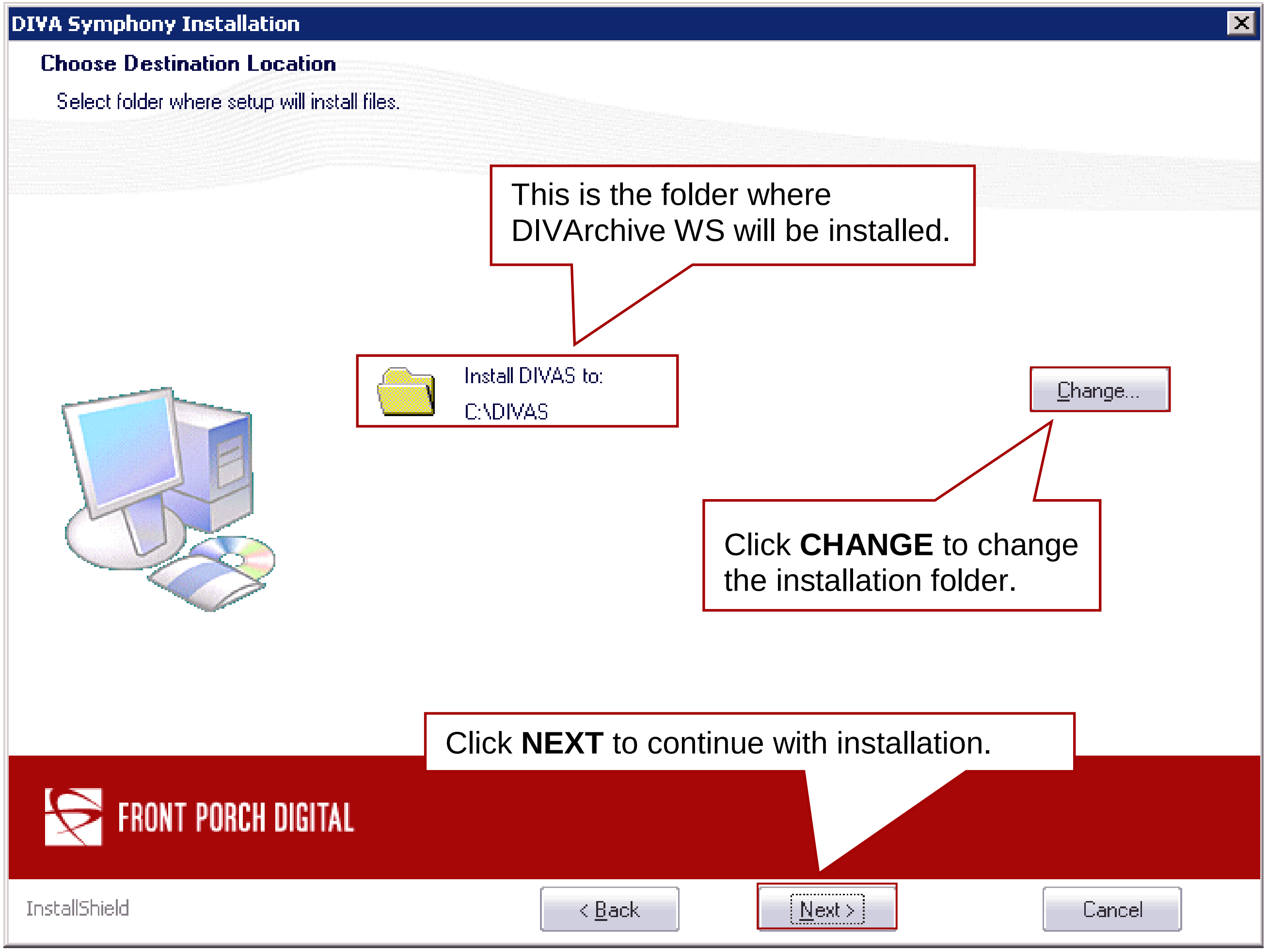
The InstallShield Wizard will install DIVArchiveWS on your computer. To continue, click Next.

Click **NEXT** to continue with installation.

4. The installer has the option to change the installation folder if desired.

Note: It is highly recommended that the default installation folder (*C:\DIVAS*) is used.

Figure 4: Choosing Destination Location



DIVA Symphony Installation

Choose Destination Location

Select folder where setup will install files.

This is the folder where DIVArchive WS will be installed.

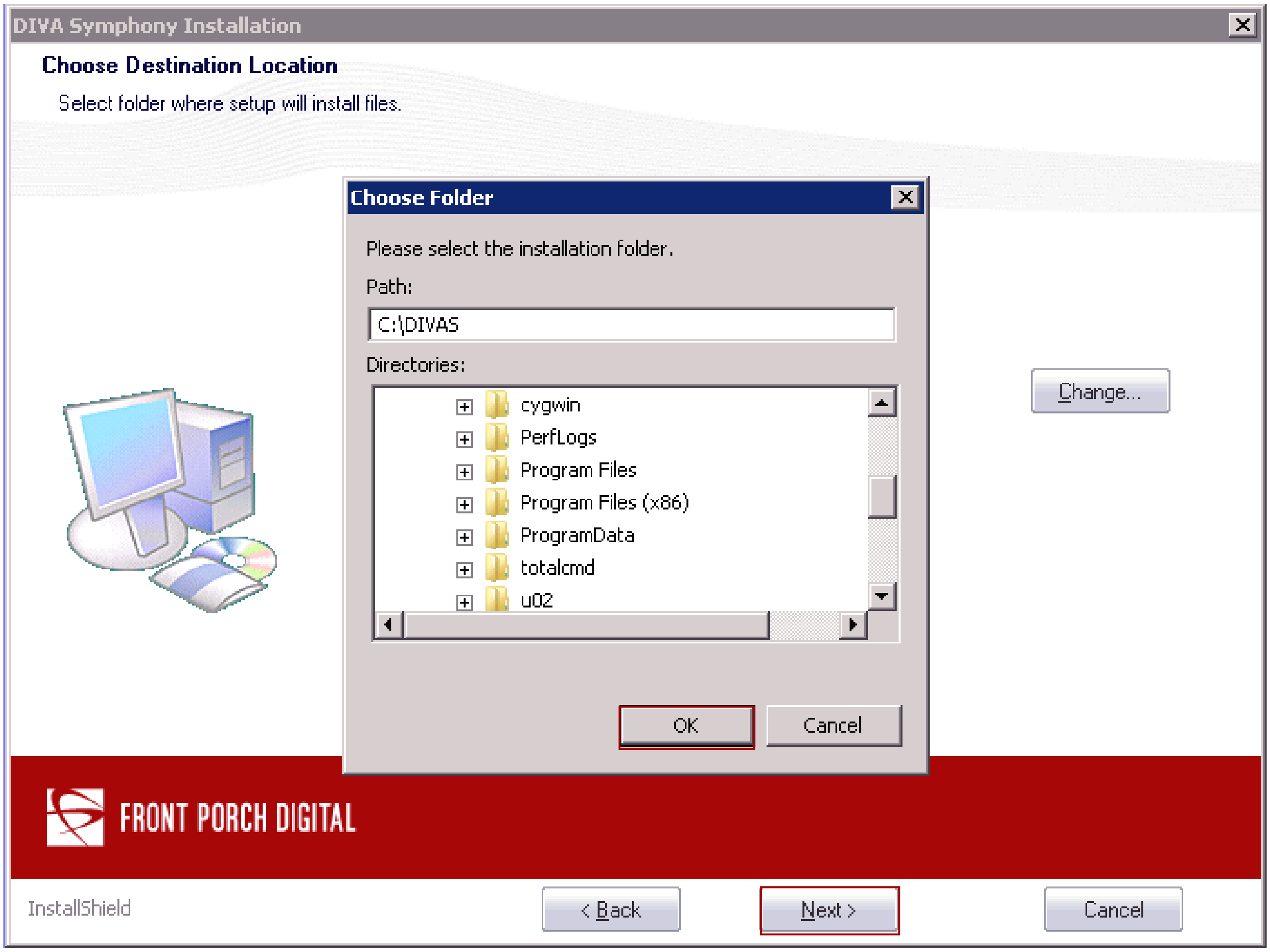
Install DIVAS to:
C:\DIVAS

Click **CHANGE** to change the installation folder.

Click **NEXT** to continue with installation.

5. Clicking the **CHANGE** button results in a dialog box allowing selection of a different installation folder (*see the figure below*).

Figure 5: Select New Installation Destination Folder Dialog



DIVA Symphony Installation

Choose Destination Location

Select folder where setup will install files.

Choose Folder

Please select the installation folder.

Path: C:\DIVAS

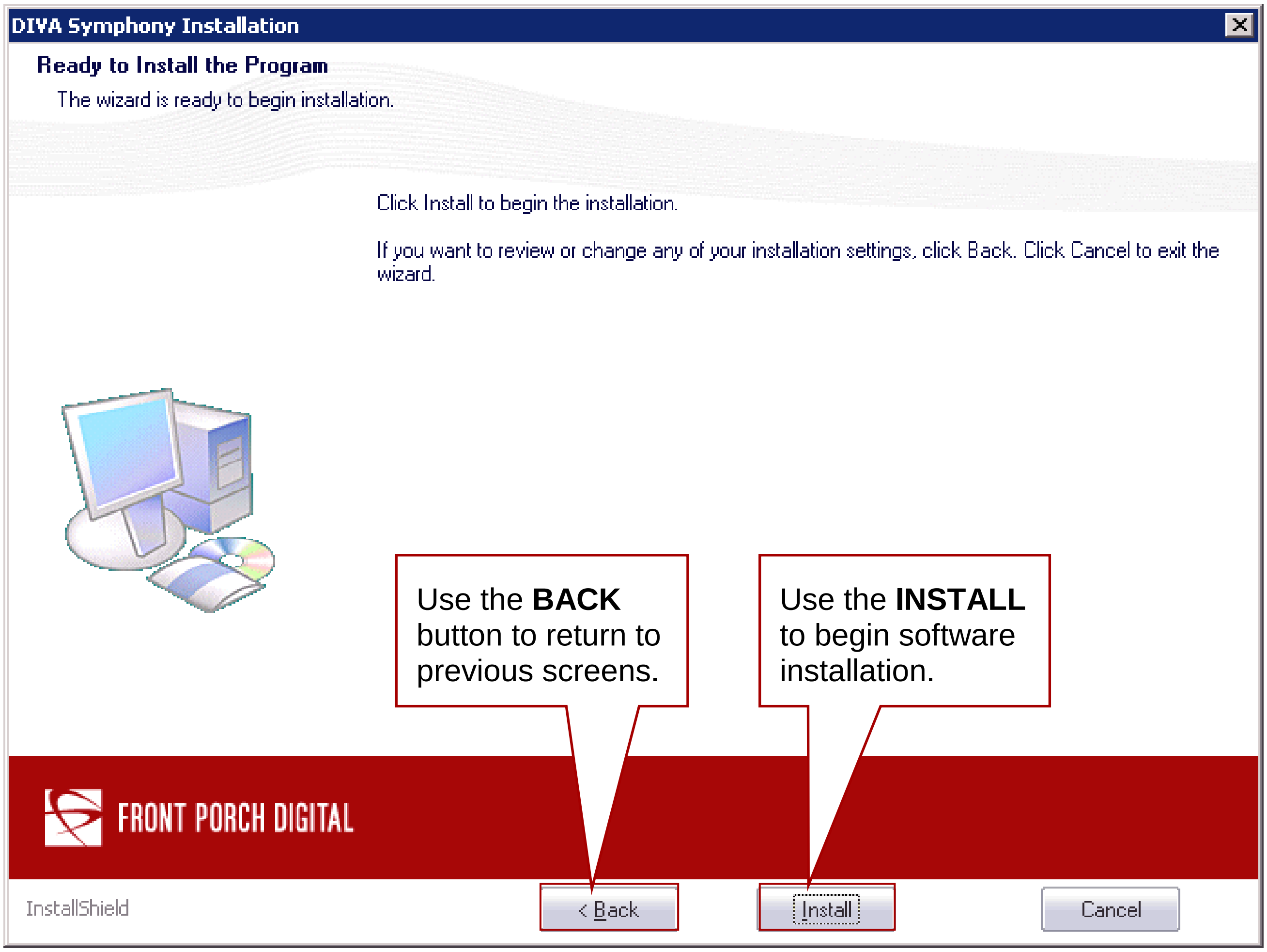
Directories:

- cygwin
- PerfLogs
- Program Files
- Program Files (x86)
- ProgramData
- totalcmd
- u02

6. Select the desired destination folder for the installation and click **OK** to use the selected folder.
7. Once returned to the **Destination Location** screen, click **NEXT** to continue the installation.

8. The DIVArchive WS Installation Program will now proceed with the actual software installation. If any parameters require modification before the software installation is performed use the **BACK** button to return to the previous screens, otherwise click **INSTALL** to begin the software installation.

Figure 6: Beginning the Actual Installation



DIVA Symphony Installation

Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

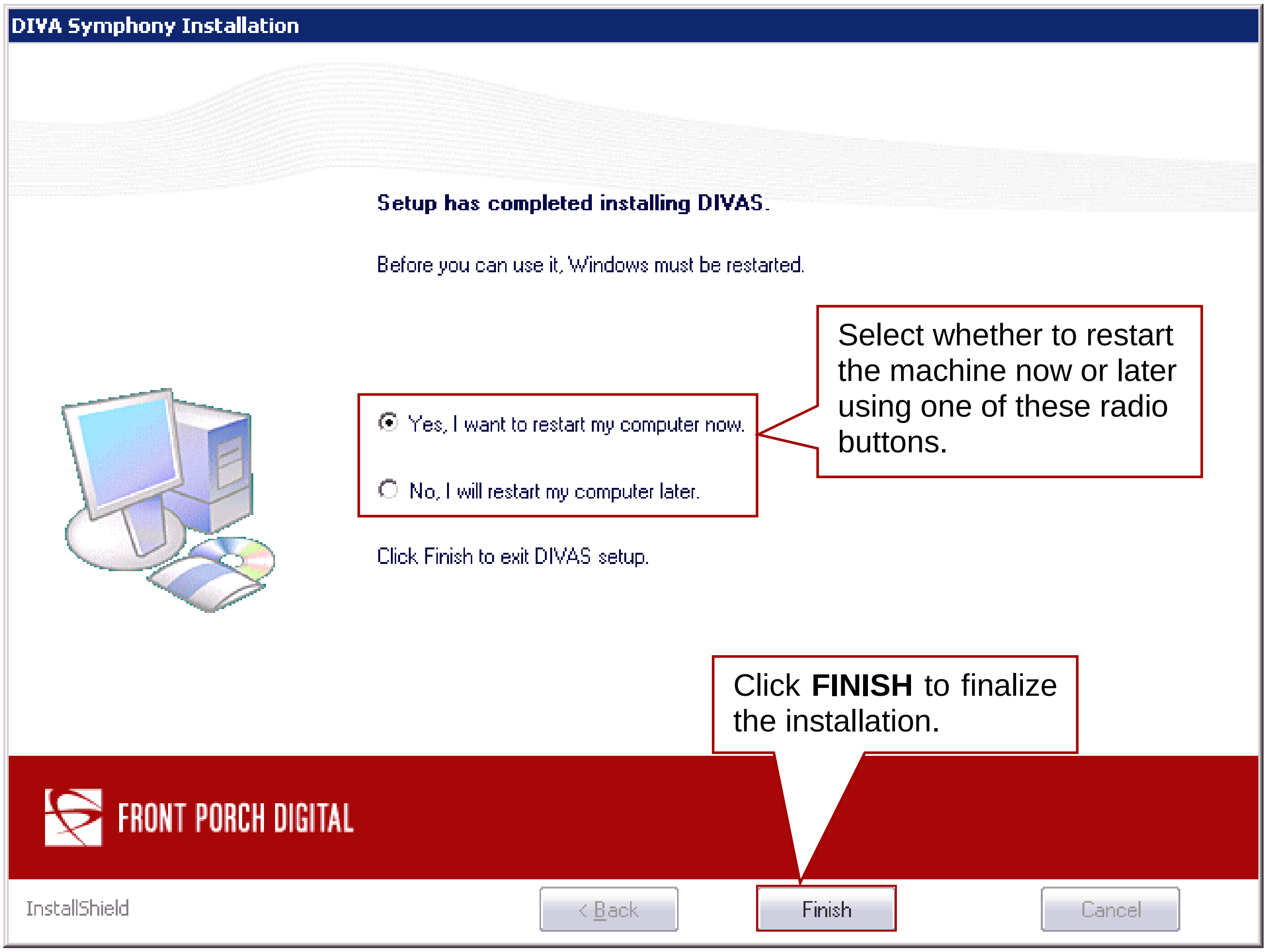
Use the **BACK** button to return to previous screens.

Use the **INSTALL** to begin software installation.

9. Once the installation process is complete, the **Installation Finished** screen will appear. Select whether to restart the machine now, or later, using the radio buttons as shown in the figure below. Click on **FINISH** to finalize the installation and exit the installer program.

10. It is recommended to restart the machine now. The new environment variables will not be visible and installation cannot proceed until the machine is restarted, or the administrator has logged out of the machine (*the administrator must log out locally, not from a remote connection*).

Figure 7: Installation is Complete



DIVA Symphony Installation

Setup has completed installing DIVAS.

Before you can use it, Windows must be restarted.

● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

Click Finish to exit DIVAS setup.

Select whether to restart the machine now or later using one of these radio buttons.

Click **FINISH** to finalize the installation.

11. After the machine has been restarted, check that the Environment Variables have been set correctly using the following commands from the Windows System Command Window:

```
C:\>set DIVAS_HOME
      DIVAS_HOME=c:\DIVAS

C:\>set CARBON_HOME
      CARBON_HOME=c:\DIVAS\application-server

C:\>set JAVA_HOME
      JAVA_HOME=c:\Program Files (x86)\Java\jre1.6.0_21\
```

12. The log file is created after the first time DIVArchive WS is run and is located in the **DIVAS_HOME\logs\divarchive_ws.trace.log** folder. The component runtime status may be printed from the log files located in the **DIVAS_HOME\<component_name>\logs\** folder with the filename **DIVAWS.status** after the status command has been executed (*Please see* WS User Commands*4.1 for more information on the status command*).

    Example:

    - **DIVAS_HOME\components\DIVArchiveWS.1.0\logs\DIVAWS.status**
    - **DIVAS_HOME\components\DIVArchiveWS.2.0\logs\DIVAWS.status**

13. Installation is completed by setting the configuration files (*as required – refer to Section 3 for more information*) and then running either:

    - **Setup.DIVArchiveWS.1.0.bat**
    - **Setup.DIVArchiveWS.2.0.bat**
    - **Setup.DIVArchiveWS.2.1.bat**
    - **Setup.all.components.bat**
        - Refer to 4.2.1.1
    - **divas.bat**
        - Refer to 4.2.1.2

2.2.1 Default Installation Folders

The default folders created during installation are:

- **DIVAS_HOME**
    - Should be defined using the **DIVAS_HOME** variable.
- **DIVAS_HOME\application-server\**
    - This should be defined using the **CARBON_HOME** variable.
    - **No changes should be made in this folder unless absolutely necessary.**
- **DIVAS_HOME\bin\**
    - This folder contains the script files used to run commands.
- **DIVAS_HOME\components\[component_name]**
    - There will be multiple folders similar to this, depending upon the components installed.
        - For example, the **DIVAS_HOME\components\DIVArchiveWS.1.0** folder contains the *DIVArchive WS 1.0* component.
- **DIVAS_HOME\tools\[tool_name]**
    - There will be multiple folders similar to this, depending upon the tools installed.

- For example, the **`DIVAS_HOME\components\DIVArchiveWSClient.1.0`** folder contains the **DIVArchive WS Client 1.0** tool.

The installation process does not add these folder names to the Windows Default Path and therefore all scripts and commands must be run either using the full path (***`DIVAS_HOME\bin\[script_name] [options]`***) or from the **`DIVAS_HOME\bin\`** folder itself.

Example:

```
DIVAS_HOME\bin\divas.bat start
```

2.2.2 Primary Folders and Scripts Installed

During installation the following primary folders and script files are created:

- **`DIVAS_HOME\`**
- **`DIVAS_HOME\bin\`**
- **`DIVAS_HOME\bin\CollectSysInfo.bat`**
  - Script used to view log information.
- **`DIVAS_HOME\bin\divas.bat`**
  - Main script used to control the system (*Refer to Section 4.1 for available options*).
- **`DIVAS_HOME\bin\DIVASHelp.txt`**
  - Help file for the **`divas.bat`** script.
- **`DIVAS_HOME\bin\Setup.DIVArchiveWS.1.0.bat`**
  - Script to configure and start **`DIVArchiveWS.1.0`** with the default configuration.
- **`DIVAS_HOME\bin\Setup.DIVArchiveWS.2.0.bat`**
  - Script to configure and start **`DIVArchiveWS.2.0`** with the default configuration.
- **`DIVAS_HOME\bin\Setup.DIVArchiveWS.2.1.bat`**
  - Script to configure and start **`DIVArchiveWS.2.1`** with the default configuration.
- **`DIVAS_HOME\bin\Setup.all.components.bat`**
  - Script to configure and start all components with the default configuration.
- **`DIVAS_HOME\bin\version.txt`**
  - Delivery package version.

2.2.3 Component Folders and Scripts Installed

During installation the following component folders and script files are created (***`xx`*** *in the filename represents the file's version number*):

- **`DIVAS_HOME\components\DIVArchiveWS.1.0\`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\bin\`**

- **`DIVAS_HOME\components\DIVArchiveWS.1.0\bin\operations.bat`**
  - All comments in **`operations.bat`** are accessible via the **`divas.bat`** script.
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\bin\version.txt`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\bundles\`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\bundles\DIVArchiveWS1.0.xx.jar`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\conf\`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\conf\conf.properties`**
  - This is the component configuration file.
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\lib\`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\lib\status-invoker-1.0.xx.jar`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\lib\divarchive-ws-common-1.0.xx.jar`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\logs\`**
  - The DIVArchive WS components status file is located in this folder.

    divarchive_ws.trace.log
  - Log files are accessed and used by the **`CollectSysInfo.bat`** script.
  - Refer to Section 4.2.5 for details.
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\references\DIVArchiveWS.xsd`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\references\DIVArchiveWS_exception.xsd`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\references\DIVArchiveWS_model.xsd`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\references\DIVArchiveWS_response.xsd`**
- **`DIVAS_HOME\components\DIVArchiveWS.1.0\references\DIVArchiveWS_SOAP.wsdl`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.0\`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.0\bin\`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.0\bin\operations.bat`**
  - All comments in **`operations.bat`** are accessible via the **`divas.bat`** script.
- **`DIVAS_HOME\components\DIVArchiveWS.2.0\bin\version.txt`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.0\bundles\`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.0\bundles\DIVArchiveWS-2.0.xx.jar`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.0\conf\`**

- **DIVAS_HOME\components\DIVArchiveWS.2.0\conf\conf.properties**
  - This is the component configuration file.
- **DIVAS_HOME\components\DIVArchiveWS.2.0\lib\**
- **DIVAS_HOME\components\DIVArchiveWS.2.0\lib\status-invoker-2.0.xx.jar**
- **DIVAS_HOME\components\DIVArchiveWS.2.0\lib\divarchive-ws-common-2.0.xx.jar**
- **DIVAS_HOME\components\DIVArchiveWS.2.0\logs\**
  - The DIVArchive WS components status file is located in this folder.

    divarchive_ws.trace.log
  - Log files are accessed and used by the **CollectSysInfo.bat** script.
  - Refer to Section 4.2.5 for details.
- **DIVAS_HOME\components\DIVArchiveWS.2.0\references\DIVArchiveWS.xsd\DIVArchiveWS.xsd**
- **DIVAS_HOME\components\DIVArchiveWS.2.0\references\DIVArchiveWS_exception.xsd**
- **DIVAS_HOME\components\DIVArchiveWS.2.0\references\DIVArchiveWS_model.xsd**
- **DIVAS_HOME\components\DIVArchiveWS.2.0\references\DIVArchiveWS_response.xsd**
- **DIVAS_HOME\components\DIVArchiveWS.2.0\references\DIVArchiveWS_SOAP.wsdl**
- **DIVAS_HOME\components\DIVArchiveWS.2.1\**
- **DIVAS_HOME\components\DIVArchiveWS.2.1\bin\**
- **DIVAS_HOME\components\DIVArchiveWS.2.1\bin\operations.bat**
  - All comments in **operations.bat** are accessible via the **divas.bat** script.
- **DIVAS_HOME\components\DIVArchiveWS.2.1\bin\version.txt**
- **DIVAS_HOME\components\DIVArchiveWS.2.1\bundles\**
- **DIVAS_HOME\components\DIVArchiveWS.2.1\bundles\DIVArchiveWS-2.0.xx.jar**
- **DIVAS_HOME\components\DIVArchiveWS.2.1\conf\**
- **DIVAS_HOME\components\DIVArchiveWS.2.1\conf\conf.properties**
  - This is the component configuration file.
- **DIVAS_HOME\components\DIVArchiveWS.2.1\lib\**
- **DIVAS_HOME\components\DIVArchiveWS.2.1\lib\status-invoker-2.0.xx.jar**

- **`DIVAS_HOME\components\DIVArchiveWS.2.1\lib\divarchive-ws-common-2.0.xx.jar`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.1\logs\`**
  - The DIVArchive WS components status file is located in this folder.

    divarchive_ws.trace.log
  - Log files are accessed and used by the **`CollectSysInfo.bat`** script.
  - Refer to Section 4.2.5 for details.
- **`DIVAS_HOME\components\DIVArchiveWS.2.1\references\DIVArchiveWS.xsd\DIVArchiveWS.xsd`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.1\references\DIVArchiveWS_exception.xsd`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.1\references\DIVArchiveWS_model.xsd`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.1\references\DIVArchiveWS_response.xsd`**
- **`DIVAS_HOME\components\DIVArchiveWS.2.1\references\DIVArchiveWS_SOAP.wsdl`**

2.3 Recommended Default Throttling Configuration

The number of simultaneously users accessing the service can be limited using **Access Throttling**. Configuration of the Data Service Throttling is accomplished using the **Throttling Wizard** using the following procedure:

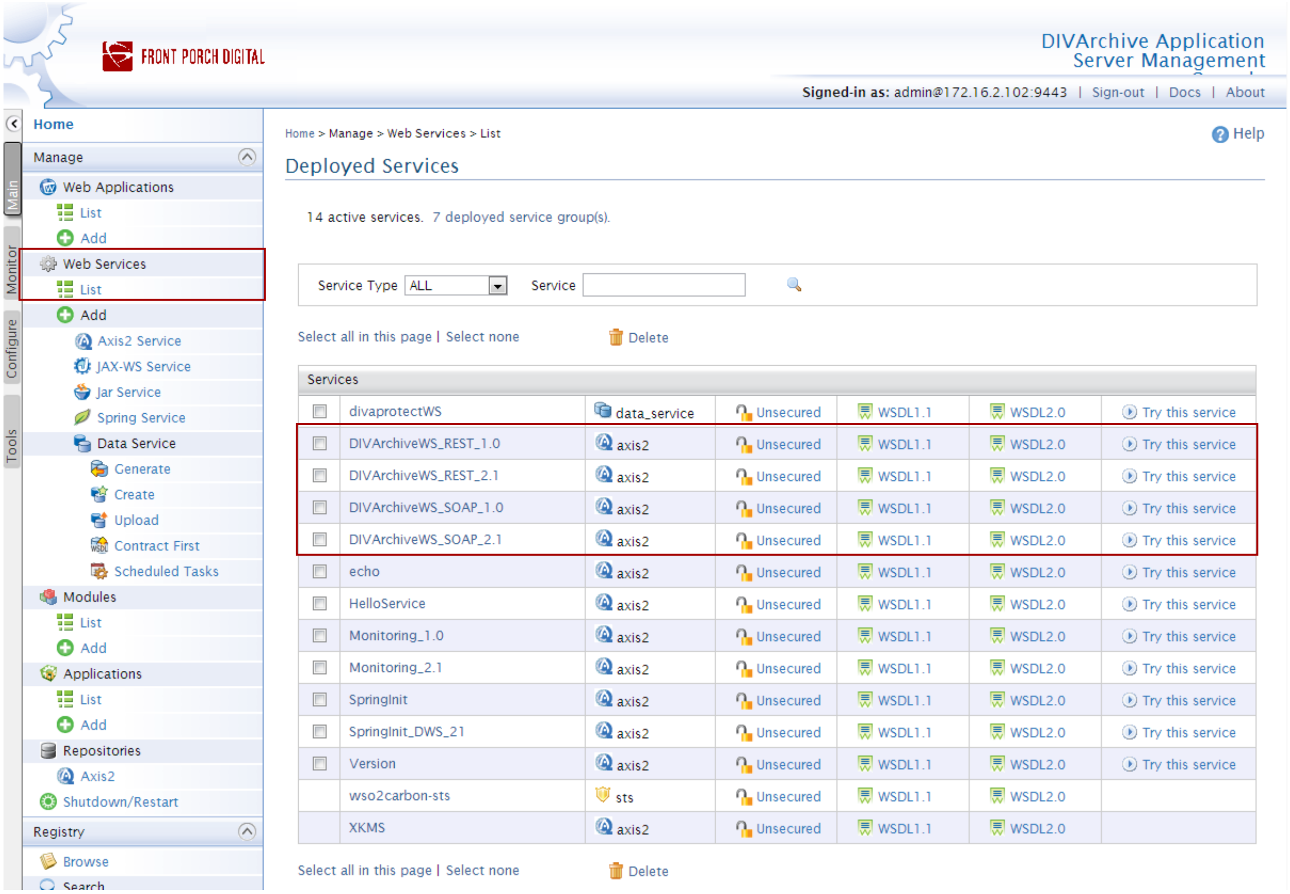
Notes:

Please refer to WSO2 Throttling Configuration Manual for more information: http://wso2.org/library/articles/wso2-throttling.

If using DIVAprotectWS, please refer to the DIVAprotect WS API User Guide for additional information.

1. Connect to the **DIVArchive Application Server Management Console**.
2. Under **Web Services** click on the **List Menu Item**.
3. Click on the **WS API** version being used.

Figure 8: Web Services – Deployed Services List



Deployed Services

14 active services. 7 deployed service group(s).

Service Type ALL Service

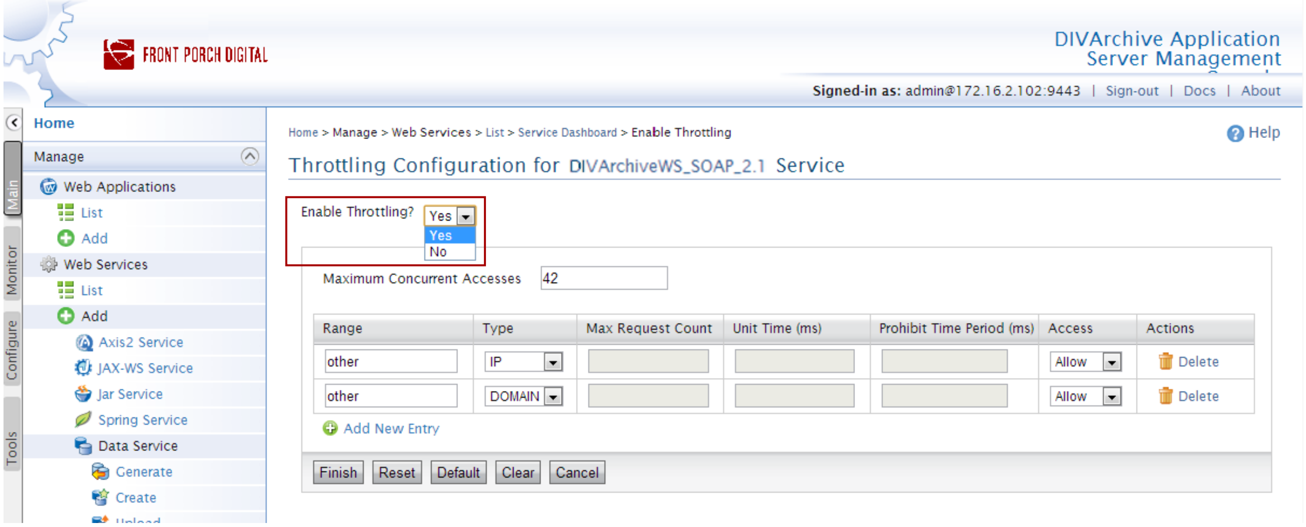
Select all in this page | Select none Delete

Services						
☐	divaprotectWS	data_service	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	DIVArchiveWS_REST_1.0	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	DIVArchiveWS_REST_2.1	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	DIVArchiveWS_SOAP_1.0	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	DIVArchiveWS_SOAP_2.1	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	echo	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	HelloService	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	Monitoring_1.0	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	Monitoring_2.1	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	SpringInit	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	SpringInit_DWS_21	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
☐	Version	axis2	Unsecured	WSDL1.1	WSDL2.0	Try this service
	wso2carbon-sts	sts	Unsecured	WSDL1.1	WSDL2.0	
	XKMS	axis2	Unsecured	WSDL1.1	WSDL2.0	

Select all in this page | Select none Delete

4. Click on **Access Throttling**.
5. Select **Yes** from the dropdown menu next to **Enable Throttling**.

Figure 9: Initial Throttling Configuration Screen



DIVArchive Application Server Management

Signed-in as: admin@172.16.2.102:9443 | Sign-out | Docs | About

Home > Manage > Web Services > List > Service Dashboard > Enable Throttling

Throttling Configuration for DIVArchiveWS_SOAP_2.1 Service

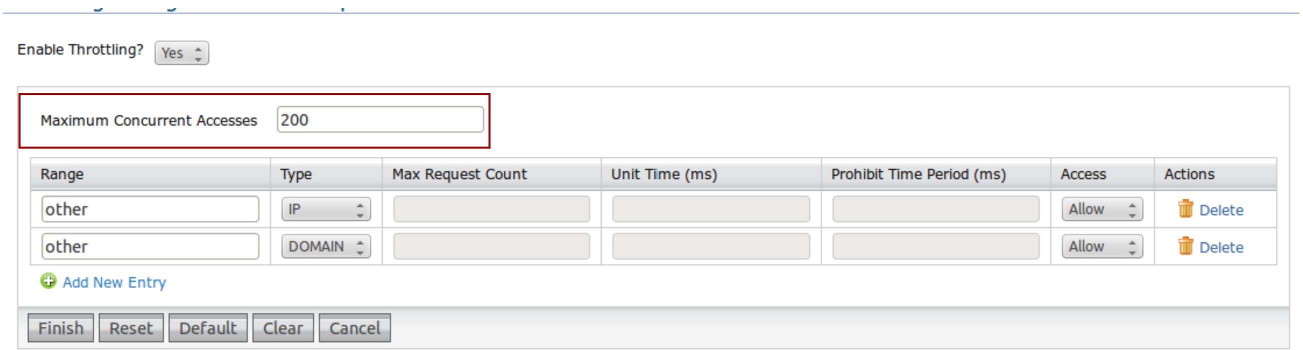
Enable Throttling? Yes (Yes / No)

Maximum Concurrent Accesses 42

Range	Type	Max Request Count	Unit Time (ms)	Prohibit Time Period (ms)	Access	Actions
other	IP				Allow	Delete
other	DOMAIN				Allow	Delete

6. The user may now configure the **Access Throttling** according to their specific requirements. When done with the configuration click one of the buttons at the bottom of the screen as described below.
7. It is highly recommended to set the maximum concurrent connections to 200 as shown in the figure below.

Figure 10: Maximum Concurrent Connections



Enable Throttling? Yes

Maximum Concurrent Accesses 200

Range	Type	Max Request Count	Unit Time (ms)	Prohibit Time Period (ms)	Access	Actions
other	IP				Allow	Delete
other	DOMAIN				Allow	Delete

Table 3: Access Throttling Button Functions

Button Name	Function
Finish	Submits the current data shown on the screen and redirects to the previous screen.
Reset	Loads the last submitted configuration.
Default	Loads the default throttle configuration. To submit the configuration after using the **Default Button**, click the **Finish Button**.
Clear	Clears all text boxes on the screen.
Cancel	Returns to the previous page without saving changes.

8. The table below identifies all areas that may be configured and a brief explanation of each. When finished configuring the **Access Throttling**, click the **Finish Button** to complete the configuration. Clicking the **Add New Entry Link** will create a new blank line to enter additional configuration items.

Table 4: Access Throttling Parameter Descriptions

Parameter	Description
Enable Throttling?	Enables or disables Access Throttling.
Maximum Concurrent Accesses	Used to control the number of requests that are served at any given moment. If throttling is enabled globally, this value will be the maximum number of requests that are served by all the services deployed in the server simultaneously. If throttling is enabled at the service level, it is the maximum number of requests for that particular service. If it is enabled at the operation level, it is the maximum requests for that particular operation.
Range	The IP address range or the domain that is restricted from accessing the service. Requests from these clients will be restricted based on the specified values.

Parameter	Description
Type	This indicates the Type of Range. Valid values are IP or DOMAIN and can be selected using the dropdown box. If the range is given as a single IP address or a range of IP addresses (*e.g. 10.100.1.30-10.100.1.60*), this should be set to read IP. If the range is given as a domain (*e.g. *.wso2.com*), this should be set to read DOMAIN. If configurations of both types are specified (*IP and DOMAIN*) the highest (*first*) priority will be given to the DOMAIN level configurations.
Access	If this is set to **Allow**, no restriction is applied for that range and all requests are allowed to be submitted as they come arrive. If this is set to **Deny**, access is completely denied for that range. When the **Access** is set to **Allow** or **Deny**, **MRC**, **UT** and **PTP** parameters are not necessary and these fields are de-activated. If **Access** is set to **Control**, the specified constraints are applied for that particular range.
Maximum Request Count **(*MRC*)**	If **Access** is set to **Control**, this will be the maximum number of requests that are served within the time interval specified by the **Unit Time** (*UT*) parameter.
Unit Time (ms) **(*UT*)**	The time period in milliseconds (*ms*) during which the maximum requests are served. This is the number specified by the **Maximum Request Count** (*MRC*). The throttle starts counting the number of units from the moment it is enabled and the number of requests served within that period.
Prohibit Time Period (ms) **(*PTP*)**	If the **Maximum Request Count** (*MRC*) is achieved before the **Unit Time** (*UT*), this is the period during which no more requests are allowed to be submitted. Setting this value alters the **Unit Time** slot. **Example :** MRC = 50, UT = 50000, PTP = 5000 In this example, if 50 requests arrive within 35000ms (*35 seconds*) within a particular time period, no more requests are taken in for another 5000ms (*PTP = 5 seconds*). This time, the UT is altered to 35000ms + 5000ms = 40000ms (*40 seconds*).

Parameter	Description
Action	Clicking on the **Delete Link** will remove only the row associated with the link that was clicked.

3 DIVArchive WS Configuration

DIVArchive WS is very simple to configure. Typical installations (*WS installed on the Oracle DIVArchive Manager Machine*) are capable of using the default configuration that is set up during installation and no further configuration is required. However, this section describes how to customize your configuration for installations that do **NOT** reside on the same machine as Oracle DIVArchive Manager.

3.1 Configuration File

The WS configuration file is located in:

```
DIVAS_HOME\components\DIVArchiveWS.2.1\conf\conf.properties
```

The configuration file is a plain text file containing the variables listed in the table below. To change the DIVArchive WS Configuration, the service must be stopped first and then restarted after the changes are complete (*Refer to Section 4.1 for the commands for starting and stopping the service*).

Table 5: DIVArchive WS **`conf.properties`** *Configuration Variables*

Variable	Parameter Definition
`DIVARCHIVE_HOST`	DIVArchive Host IP Address.
`DIVARCHIVE_PORT`	DIVArchive Port Number.
`DIVARCHIVE_MAX_SIMULTANEOUS_REQUESTS`	Number of maximum simultaneous requests.
`DIVARCHIVE_RESPONSE_TIMEOUT_IN_MINUTES`	Response Timeout in minutes. The default is 10 minutes.
`DIVARCHIVE_USER_NAME`	DIVArchive Login Name. *This is only required when using the Oracle DIVAnet Access Gateway.*
`DIVARCHIVE_PASSWORD`	DIVArchive Password. *This is only required when using the Oracle DIVAnet Access Gateway.*
`DIVARCHIVE_APPLICATION_NAME`	Application Name used to connect with the DIVArchive System. *This is only required when using the Oracle DIVAnet Access Gateway and for security purposes.*

Variable	Parameter Definition
DIVARCHIVE_SESSION_WATCHER_INTERVAL_IN_MINUTES	The time interval (*in minutes*) indicating how often the DIVArchive Session should be checked and retried in case of disconnection.
SESSION_STORAGE_TIMEOUT_IN_MINUTES	The time interval (*in minutes*) indicating how long the system will attempt to reconnect a failed connection. The Session expires if it has used more than the given time interval.
SESSION_STORAGE_SWEEPER_INTERVAL_IN_MINUTES	The time interval (*in minutes*) indicating how often the Session Sweeper should dispose of expired sessions.
MONITORING_INTERVAL_IN_MINUTES	The time interval (*in minutes*) indicating how often the monitoring service writes data to the log file.
MONITORING_STATUS_FILE_RELATIVE_PATH	The Relative Path to the system status information file.
MONITORING_HTTP_ENDPOINT	URL of the monitoring web service.
LOGGING_DIRECTORY	Relative Path to the logging directory. This is used by the *LoggerManager*.
LOGGING_ROOT_LEVEL	This tells the *LoggerManager* which level of detail is required for the Root Path. Levels can be **DEBUG**, **INFO**, **WARN**, **ERROR** or **FATAL**.
LOGGING_TRACE_LEVEL	This tells the *LoggerManager* which level of detail is required for system tracing. Levels can be **DEBUG**, **INFO**, **WARN**, **ERROR** or **FATAL**.

Variable	Parameter Definition
`LOGGING_SERVICE_LEVEL`	This tells the *LoggerManager* which level of detail is required for the system service. Levels can be **`DEBUG`**, **`INFO`**, **`WARN`**, **`ERROR`** or **`FATAL`**.
`LOGGING_MAXFILESIZE`	The maximum size the log file can grow to before it is compressed and a new log file is started. The size should be specified using **`KB|MB`**.
`LOGGING_LIFETIME`	The time (*in hours*) that specifies when all files (*trace, service and .zip*) older than this will be removed.

3.2 Changing the WS IP Address (if required)

Changing the WS IP Address is only necessary in the event that the installation machine has multiple network cards and/or IP Addresses active.

Note: It is recommended that a copy of the `carbon.xml` file be made (*for backup purposes*) before editing the original.

Warning: Nothing should be changed in the `carbon.xml` file other than the Hostname!

To change the IP Address in the configuration:

1. Use Notepad (*or any plain text editor*) to edit the **`carbon.xml`** configuration file
   - The configuration file is located in the **`DIVAS_HOME\application-server\repository\conf\`** folder.
2. Set the value of the **`HostName`** element to the IP Address desired to appear in the WSDL files.

3.3 Limiting the Number of Log Files

The DIVArchive WS Logging System has no limit to the number of log files created. By default, DIVArchive WS creates a new log file every time the previous log file is full (*refer to **Error! Reference source not found.** for information on log file parameters*).

To avoid issues with a large number of log files being created, the maximum number of log files permitted may be configured as follows:

1. Open the **`DIVA_HOME\application-server\repository\conf\wrapper.conf`** file.
2. Find the **`wrapper.logfile.maxfiles`** parameter and set it to the necessary number of log files.
   a. wrapper.logfile.maximum=20
3. Save the file.

In the example in Step 2.a (*above*), every time the number of maximum log files is reached (*i.e. 20 in the example*), DIVArchive WS will begin to overwrite the existing log files. It is important to make certain that the maximum number of files is adequate for the system this parameter is being configured on, yet not so large that it will cause disk space issues.

4 Operations

4.1 WS User Commands

The DIVArchive WS Application supports using a command line interface to perform system functions. The command line interface is accessed through the Windows Command Window (***START, RUN, CMD***) and must be initiated as an administrative user because administrator permissions are required to run the script files.

The WS Service Commands can be executed using the batch script file named **divas.bat**, while individual components have their own scripts and commands. The service may also be started and stopped using the Windows Service Panel.

The syntax for initiating commands using the batch script is:

```
DIVAS_HOME\bin\divas.bat command [options]
```

Example:

DIVAS_HOME\bin\divas.bat start (*This command starts the WS Service*)

The table below identifies the available commands and their effect on the WS Application.

Note: When deploying or un-deploying components, make sure the service has been stopped prior to running the command. The service should be restarted after the executed command has completed successfully.

*Table 6: **divas.bat** Script Command Options*

Command	Definition
install [service name]	Installs the DIVArchive Application Service. **Examples:** **DIVAS_HOME\bin\divas.bat install** **DIVAS_HOME\bin\divas.bat install DIVApplicationServer**
uninstall [service name]	Uninstalls the DIVArchive Application Service. **Examples:** **DIVAS_HOME\bin\divas.bat uninstall** **DIVAS_HOME\bin\divas.bat uninstall DIVApplicationServer**
start [service name]	Starts the DIVArchive Application Server as a Windows Service. It takes approximately 60 seconds to start the service. **Examples:** **DIVAS_HOME\bin\divas.bat start** **DIVAS_HOME\bin\divas.bat start DIVApplicationServer**

Command	Definition

`stop [service name]`	Stops the DIVArchive Application Server Windows Service.
	Examples:
	`DIVAS_HOME\bin\divas.bat stop`
	`DIVAS_HOME\bin\divas.bat stop DIVApplicationServer`

`restart [service name]`	Restarts the DIVArchive Application Server Windows Service.
	Example:
	`DIVAS_HOME\bin\divas.bat restart`
	`DIVAS_HOME\bin\divas.bat restart DIVApplicationServer`

`deploy [component folder name]`	Deploys a component to the DIVArchive Application Server. Stop the service first and then restart it after deploying the component. **Example:** `DIVAS_HOME\bin\divas.bat deploy DIVArchiveWS.1.0`
`deploy all`	Deploys all available components to the DIVArchive Application Server. Stop the service first and then restart it after deploying the components. **Example:** `DIVAS_HOME\bin\divas.bat deploy all`
`undeploy [component folder name]`	Removes a component from the DIVArchive Application Server. Stop the service first and then restart it after removing the component. **Example:** `DIVAS_HOME\bin\divas.bat undeploy DIVArchiveWS.1.0`
`undeploy all`	Remove all available components from the DIVArchive Application Server. Stop the service first and then restart it after removing the components. **Example:** `DIVAS_HOME\bin\divas.bat undeploy all`
`status [component folder name]`	Displays the detailed status of the component. **Example:** `DIVAS_HOME\bin\divas.bat status DIVArchiveWS.1.0`
`status all`	Displays the status of all components. **Example:** `DIVAS_HOME\bin\divas.bat status all`

Command	Definition
`version`	Displays the version of the DIVArchive Application Server and versions of all components. **Example:** `DIVAS_HOME\bin\divas.bat version`
`help`	Displays help information and then exits. **Example:** `DIVAS_HOME\bin\divas.bat help`

Using the `DIVAS_HOME\bin\divas.bat status` command allows viewing of a wide variety of status and usage information including:

- Overall Status
- Start Time
- DIVArchive Host
- DIVArchive Port
- Occupied Memory
- State of the DIVArchive Connection.
- State of all Client Sessions.

4.2 WS User Operations

4.2.1 Configuring and Starting the DIVArchive WS Service

4.2.1.1 Using Setup.DIVArchiveWS.1.0.bat Script

Using the **`Setup.DIVArchiveWS.1.0.bat`** script is the easiest way to install, configure, and check DIVArchive WS 1.0.x. This batch file does not accept any additional parameters and executes the following actions in sequence:

1. Deploys the DIVArchive WS 1.0.x Bundle.
2. Installs the DIVArchive WS as a Windows Service.
3. Starts the DIVArchive WS Service.
4. Executes the **`status`** command and prints out a detailed status report of the installation and started service.

4.2.1.2 Using Setup.DIVArchiveWS.2.0.bat Script

Using the **`Setup.DIVArchiveWS.2.0.bat`** script is the easiest way to install, configure, and check DIVArchive WS 2.0.x. This batch file does not accept any additional parameters and executes the following actions in sequence:

1. Deploys the DIVArchive WS 2.0.x bundle.
2. Installs the DIVArchive WS as a Windows Service.
3. Starts the DIVArchive WS Service.
4. Executes the **`status`** command and prints out a detailed status report of the installation and started service.

4.2.1.3 Using Setup.DIVArchiveWS.2.1.bat Script

Using the **`Setup.DIVArchiveWS.2.1.bat`** script is the easiest way to install, configure, and check DIVArchive WS 2.1.x. This batch file does not accept any additional parameters and executes the following actions in sequence:

1. Deploys the DIVArchive WS 2.1.x bundle.
2. Installs the DIVArchive WS as a Windows Service.
3. Starts the DIVArchive WS Service.
4. Executes the **`status`** command and prints out a detailed status report of the installation and started service.

4.2.1.4 Using the Setup.all.components.bat Script

Using the **`Setup.all.components.bat`** script is the easiest way to install, configure, and check all DIVArchive WS versions. This batch file does not accept any additional parameters and executes the following actions in sequence:

1. Deploys all of the components.
2. Installs the DIVArchive WS as a Windows Service.
3. Starts the DIVArchive WS Service.
4. Executes the **`status`** command and prints out a detailed status report of the installation and started service.

4.2.1.5 Using the `divas.bat` Script

The DIVArchive WS Service can be created and started using the **`divas.bat`** script by performing the following actions:

1. Deploy the service by executing the following command from the console:
   - **`DIVAS_HOME\bin\divas.bat deploy DIVArchiveWS.1.0`**
   - **`DIVAS\bin\divas.bat deploy DIVArchiveWS.2.0`**
   - **`DIVAS\bin\divas.bat deploy all`**
2. Install the **`DIVApplicationServer`** service:
   - **`DIVAS_HOME\bin\divas.bat install`**
3. Start the service:
   - **`DIVAS_HOME\bin\divas.bat start`**

4.2.2 Restarting DIVArchive WS

The restart procedure consists of shutting down the service and then starting it again using the *restart* command:

```
DIVAS_HOME\bin\divas.bat restart DIVApplicationServer
```

If the service was installed but not running, it will only be started (*not shut down first*). If the service was not installed then an error message will be displayed indicating that the service is not installed.

4.2.3 Shutting Down DIVArchive WS

When the service is running, it can be shut down using the *stop* command:

```
DIVAS_HOME\bin\divas.bat stop DIVApplicationServer
```

4.2.4 Checking the DIVArchive WS Status

Executing the `DIVAS_HOME\bin\divas.bat status` command creates these files and includes the most current information:

- `DIVAS_HOME\components\DIVArchiveWS.1.0\logs\DIVAWS.status`
- `DIVAS_HOME\components\DIVArchiveWS.2.0\logs\DIVAWS.status`
- `DIVAS_HOME\components\DIVArchiveWS.2.1\logs\DIVAWS.status`

The status of the DIVArchive WS Service can be checked at any time using the `status` command:

```
DIVAS_HOME\bin\divas.bat status [component_name]
```

4.2.5 Collecting DIVArchive WS Logs

The DIVArchive WS log files are located in the `DIVAS_HOME\logs\` folder and are accessed using the `CollectSysInfo.bat` script:

```
DIVAS_HOME\bin\CollectSysInfo.bat
```

Note: Information on usage of this tool can be found in the Customer Information Collection Tool User Guide. Logs and traces are produced separately per bundle. The `CollectSysInfo.bat` collection tool collects logs from both places.

4.2.6 Upgrading DIVArchive WS

Upgrading the DIVArchive WS installation uses the same basic procedure used to upgrade DIVArchive. The DIVArchive WS Service must be stopped before performing the upgrade and then restarted after the upgrade process is complete.

1. Stop the DIVArchive WS Application Service.
   - `DIVAS_HOME\bin\divas.bat stop`
2. Stop the DIVAS Application Server.
   - `divas.bat stop`
3. Uninstall Windows Services
   - `divas.bat uninstall`
4. Save the configuration file for your references:
   - `DIVAS_HOME /DIVArchiveWS1.0/conf/conf.properties`
   - `DIVAS_HOME /DIVArchiveWS2.0/conf/conf.properties`
5. Remove or rename the `DIVAS_HOME` directory.
6. Perform a fresh installation.
7. Follow the standard installation procedure using the configuration file stored for references; configure one or two bundles as necessary.

The upgrade process is now complete and the DIVArchive WS Service is running.

5 DIVArchive WS Testing and Clients

5.1 *Using the WSO2 TryIt Service*

There are two ways to determine if WS and the WS Service are operational. The first method is by using the **`status`** command:

```
DIVAS_HOME\bin\divas.bat status
```

Once the status is determined (*or not determined*) via the **`status`** command, the next step is to use the *TryIt Service*.

The **WSO2 TryIt Service** is a GUI used to try each API command and verify that it is functioning properly. The GUI has a SOAP-like look and feel making it very easy to accommodate any complexity level of XML Schema, and offers support for Schema and WSDL importing.

TryIt has the ability to position Request and Response views either vertically or horizontally and the available operations are listed in alphabetical order making them easy to locate. The ability to add frequently invoked operations to a **Priority Operations Group** allows the specified operations to appear at the top of the **Operations List** so they may be located quickly for testing.

The TryIt Service can be reached using a web browser at the following locations (*these links only work if they are opened on the same host that DIVAS is installed on*):

- http://localhost:9763/services/DIVArchiveWS_SOAP_1.0?tryit
- http://localhost:9763/services/DIVArchiveWS_SOAP_2.0?tryit
- http://localhost:9763/services/DIVArchiveWS_SOAP_2.1?tryit
- http://localhost:9763/services/DIVArchiveWS_REST_1.0?tryit
- http://localhost:9763/services/DIVArchiveWS_REST_2.0?tryit
- http://localhost:9763/services/DIVArchiveWS_REST_2.1?tryit

Notes:

- The **WSO2 TryIt program currently only works with REST. For SOAP testing please use the SOAPUI Tool.**
- **When using the TryIt Service, `test` can be used as a valid session code if `RegisterClient` will not be run prior to running another command.**

5.2 TryIt Service Limitations

The following limitations and recommendation should be adhered to when using the TryIt interface:

- **Google Chrome:** works fine (*version 15.X*) and is recommended for testing.
- **FireFox:** in some versions only SOAP is supported and REST is not supported.
- **MS Internet Explorer:** usability issues.
- **Safari:** usability issues.
- The TryIt Service currently only works with REST. For SOAP testing please use the SOAPUI Tool.

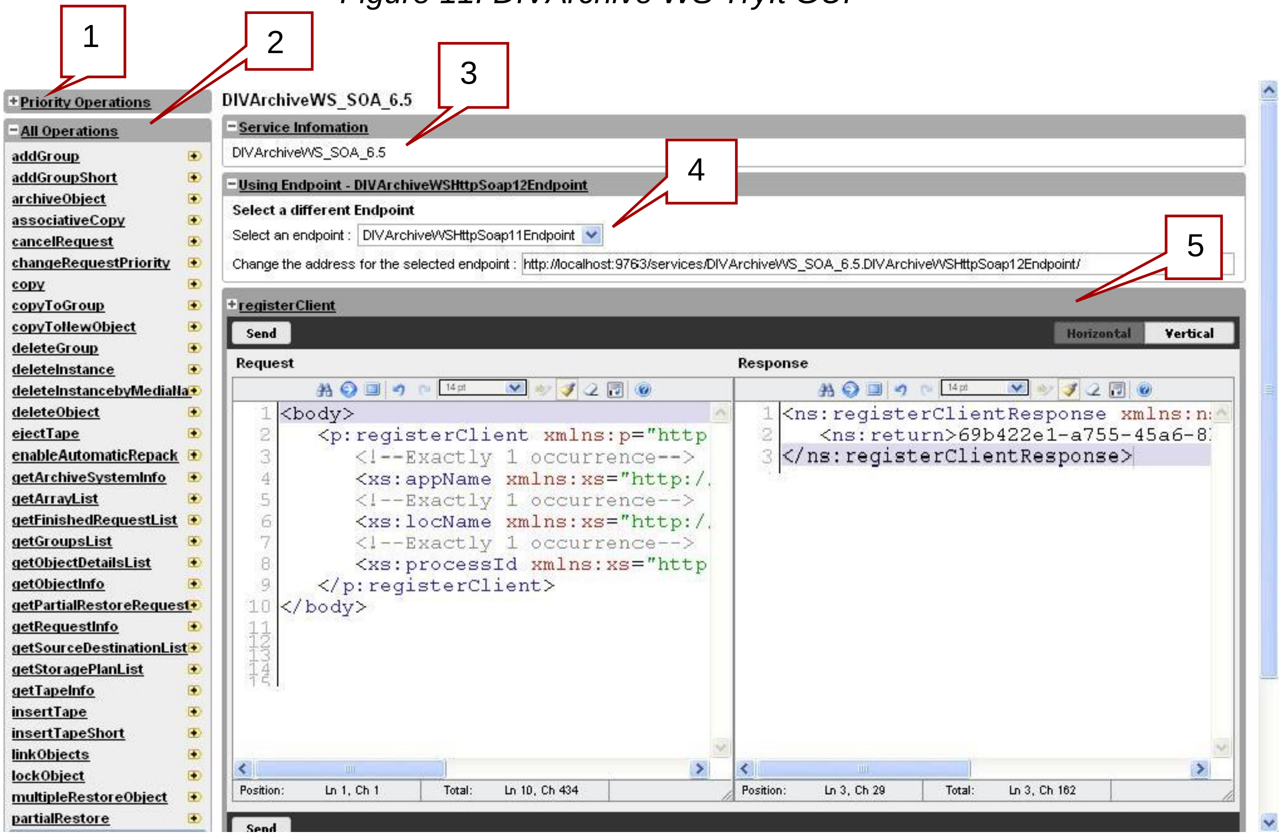
Note: Due to limitations in the current release of the TryIt service, the Send button may require to be pressed twice. However, following the directions below may cause the issues to disappear:

- **Toggle `Syntax Highlight` on/off**
- **Toggle `Word Wrap Mode` on/off**

5.3 TryIt Service Operations

Note: Operations using the TryIt Service are identical for both the SOAP and REST interfaces. However at this time the WSO2 TryIt Tool is not 100% compatible with SOAP due to issues with their code (*not related to DIVArchive*).

Figure 11: DIVArchive WS TryIt GUI



DIVArchiveWS_SOA_6.5

Select an endpoint : DIVArchiveWSHttpSoap11Endpoint

Change the address for the selected endpoint : http://localhost:9763/services/DIVArchiveWS_SOA_6.5.DIVArchiveWSHttpSoap12Endpoint/

```
<body>
   <p:registerClient xmlns:p="http
      <!--Exactly 1 occurrence-->
      <xs:appName xmlns:xs="http:/.
      <!--Exactly 1 occurrence-->
      <xs:locName xmlns:xs="http:/.
      <!--Exactly 1 occurrence-->
      <xs:processId xmlns:xs="http
   </p:registerClient>
</body>
```

Position: Ln 1, Ch 1 Total: Ln 10, Ch 434

```
<ns:registerClientResponse xmlns:n
   <ns:return>69b422e1-a755-45a6-8
</ns:registerClientResponse>
```

Position: Ln 3, Ch 29 Total: Ln 3, Ch 162

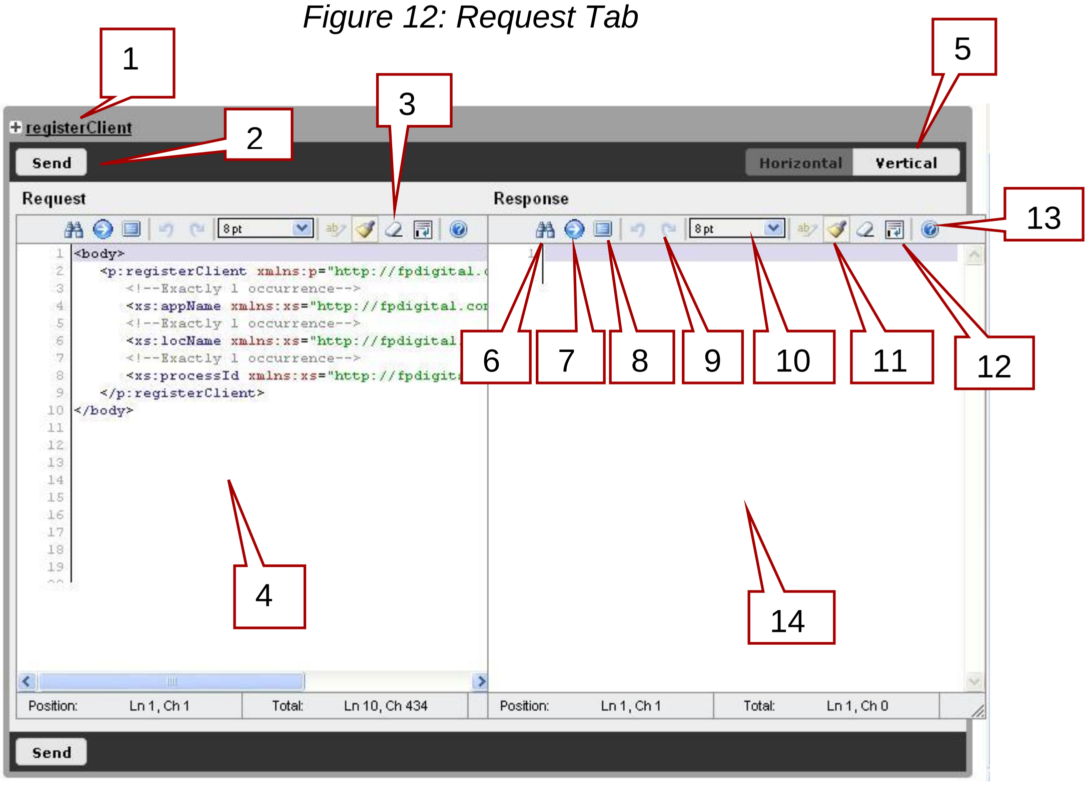
All elements of the TryIt interface are contained within the five tabs marked on the figure above. Each tab can be expanded or collapsed using the corresponding "+" or "-" icon located in the upper left corner of the tab.

The tabs designated in the figure above are as follows:

1. **Priority Operations Tab**
   - This tab contains a list of requests selected by the user as **Frequently Used** from the main list of all operations.
2. **All Operations Tab**
   - This tab contains the list of all requests available. Requests are shown in alphabetical order. You can add a request from this tab to the **Priority Operations** *Tab* by clicking on the "+" sign to the right of the operation's name.
3. **Service Information Tab**
   - This tab contains information about the service.

4. **Endpoint Tab**
    - This tab contains information about the endpoint which is currently being used to submit requests and allows changing the endpoint in use by selecting another one from the drop-down list, or by modifying it manually in the text box below the drop-down list.
5. **Selected Request Tab**
    - This tab shows the request selected by the user. Initially it shows the first request (*in alphabetical order*) from the **All Operations** *Tab*.

Figure 12: Request Tab



```
<body>
    <p:registerClient xmlns:p="http://fpdigital.c
       <!--Exactly 1 occurrence-->
       <xs:appName xmlns:xs="http://fpdigital.cor
       <!--Exactly 1 occurrence-->
       <xs:locName xmlns:xs="http://fpdigital.
       <!--Exactly 1 occurrence-->
       <xs:processId xmlns:xs="http://fpdigita
    </p:registerClient>
</body>
```

Position: Ln 1, Ch 1 Total: Ln 10, Ch 434

Position: Ln 1, Ch 1 Total: Ln 1, Ch 0

The *Request Tab* elements are described as follows:

1. **Request Name**
    - The name of the currently selected request.
2. **Send Request Button**
    - Submits a request.
3. **Toggle Word Wrapping Mode**

    - Enables XML text word wrapping in the Request/Response windows.
4. **Request Text**

    - XML for the current request.
5. **Select Layout Type**

    - Toggles vertical or horizontal layout of the Request/Response windows. The horizontal layout is shown in the figure.

6. **Find**

   - Displays a typical *Search* dialog. This is useful for large XML files and supports regular expressions.

7. **Go to Line Number**

   - Displays a typical *Go to Line* dialog.

8. **Full Screen**

   - This causes the corresponding text pane in which this option is selected to take all of the space available in the browser.

9. **Undo/Redo**

   - Typical *Undo/Redo* functionality for text editing.

10. **Font Size**

    - Sets the text font size in the Response and Request windows.

11. **Toggle Syntax Highlight On/Off**

    - Toggles the color highlighting of the XML syntax (*on or off*).

12. **Reset Highlight**

    - Allows resetting of the highlighting in case the highlight is mixed because of a browser error.

13. **About**

    - Shows a dialog with useful keyboard shortcuts, version information, etc.

14. **Response Text**

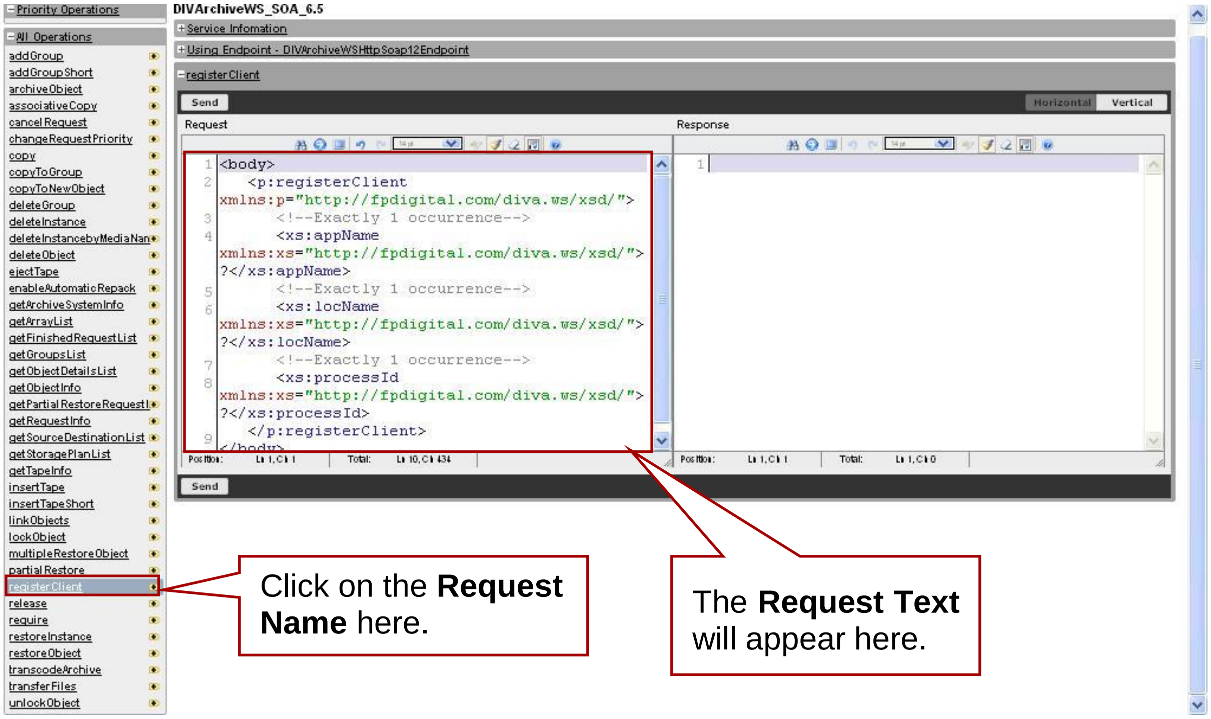
    - After the request is submitted, the XML response is displayed here.

5.3.1 Scenario 1: Submitting a `registerClient` Request

Note: Word Wrap Mode has been selected in the following examples. See the previous section for information on Word Wrap Mode.

1. Select the **`registerClient`** request from the **All Operations Tab** by clicking it.

Figure 13: Selecting a Request for Test



```
<body>
    <p:registerClient
xmlns:p="http://fpdigital.com/diva.ws/xsd/">
        <!--Exactly 1 occurrence-->
        <xs:appName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">
?</xs:appName>
        <!--Exactly 1 occurrence-->
        <xs:locName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">
?</xs:locName>
        <!--Exactly 1 occurrence-->
        <xs:processId
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">
?</xs:processId>
    </p:registerClient>
</body>
```

Click on the **Request Name** here.

The **Request Text** will appear here.

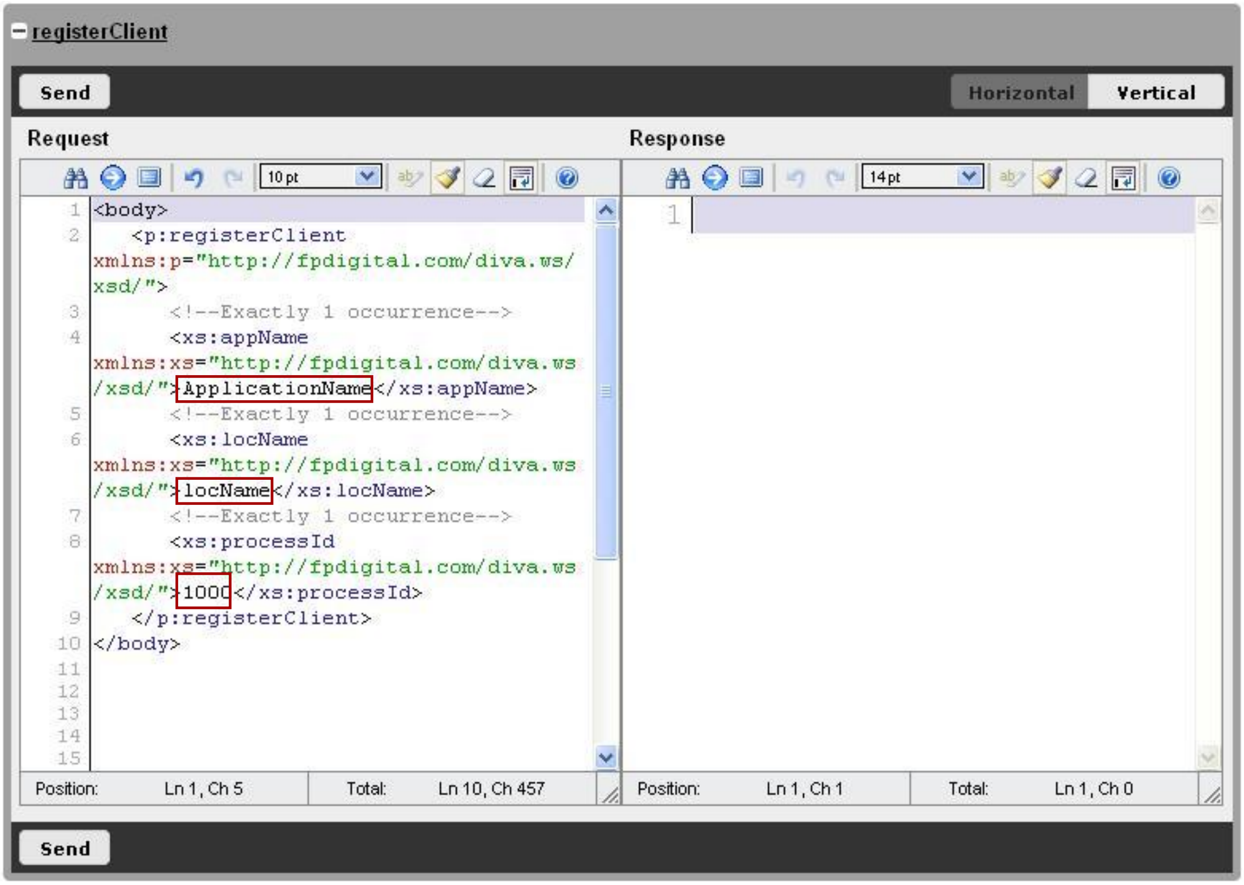
2. Fill in the correct values for the request directly in the **Request Text** window.

   Note that TryIt puts a "?" by default as the initial value for the Request Parameters. Replace all "?" symbols with the desired values.

   For the **`registerClient`** request, the following parameters must be given a value as shown in the figure below:

   - **`appName`**
   - **`locName`**
   - **`processId`**

Figure 14: These Parameters Require Values for testing ***`registerClient`***



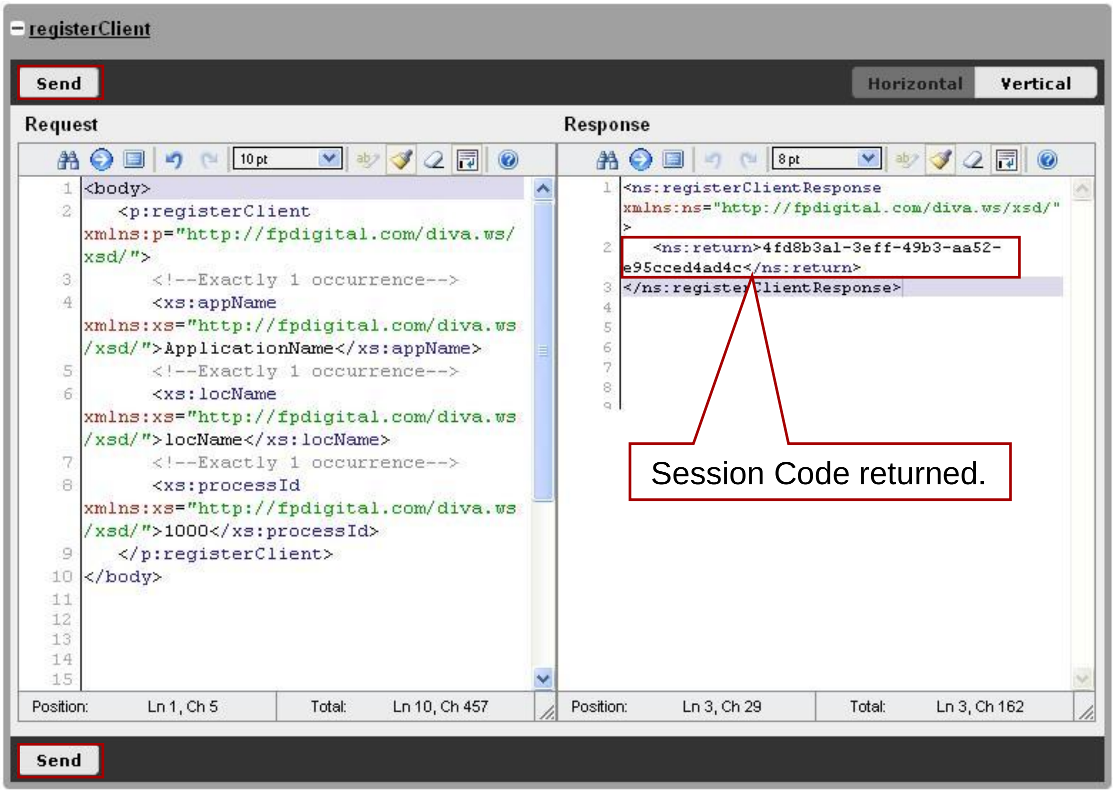
```
<body>
   <p:registerClient xmlns:p="http://fpdigital.com/diva.ws/xsd/">
      <!--Exactly 1 occurrence-->
      <xs:appName xmlns:xs="http://fpdigital.com/diva.ws/xsd/">ApplicationName</xs:appName>
      <!--Exactly 1 occurrence-->
      <xs:locName xmlns:xs="http://fpdigital.com/diva.ws/xsd/">locName</xs:locName>
      <!--Exactly 1 occurrence-->
      <xs:processId xmlns:xs="http://fpdigital.com/diva.ws/xsd/">1000</xs:processId>
   </p:registerClient>
</body>
```

Position: Ln 1, Ch 5 Total: Ln 10, Ch 457

Position: Ln 1, Ch 1 Total: Ln 1, Ch 0

3. Click the *Send* button. The response will appear with a *Session Code* value.

Figure 15: Response Containing the Session Code



Request:

```
<body>
   <p:registerClient xmlns:p="http://fpdigital.com/diva.ws/xsd/">
      <!--Exactly 1 occurrence-->
      <xs:appName xmlns:xs="http://fpdigital.com/diva.ws/xsd/">ApplicationName</xs:appName>
      <!--Exactly 1 occurrence-->
      <xs:locName xmlns:xs="http://fpdigital.com/diva.ws/xsd/">locName</xs:locName>
      <!--Exactly 1 occurrence-->
      <xs:processId xmlns:xs="http://fpdigital.com/diva.ws/xsd/">1000</xs:processId>
   </p:registerClient>
</body>
```

Response:

```
<ns:registerClientResponse xmlns:ns="http://fpdigital.com/diva.ws/xsd/">
   <ns:return>4fd8b3a1-3eff-49b3-aa52-e95cced4ad4c</ns:return>
</ns:registerClientResponse>
```

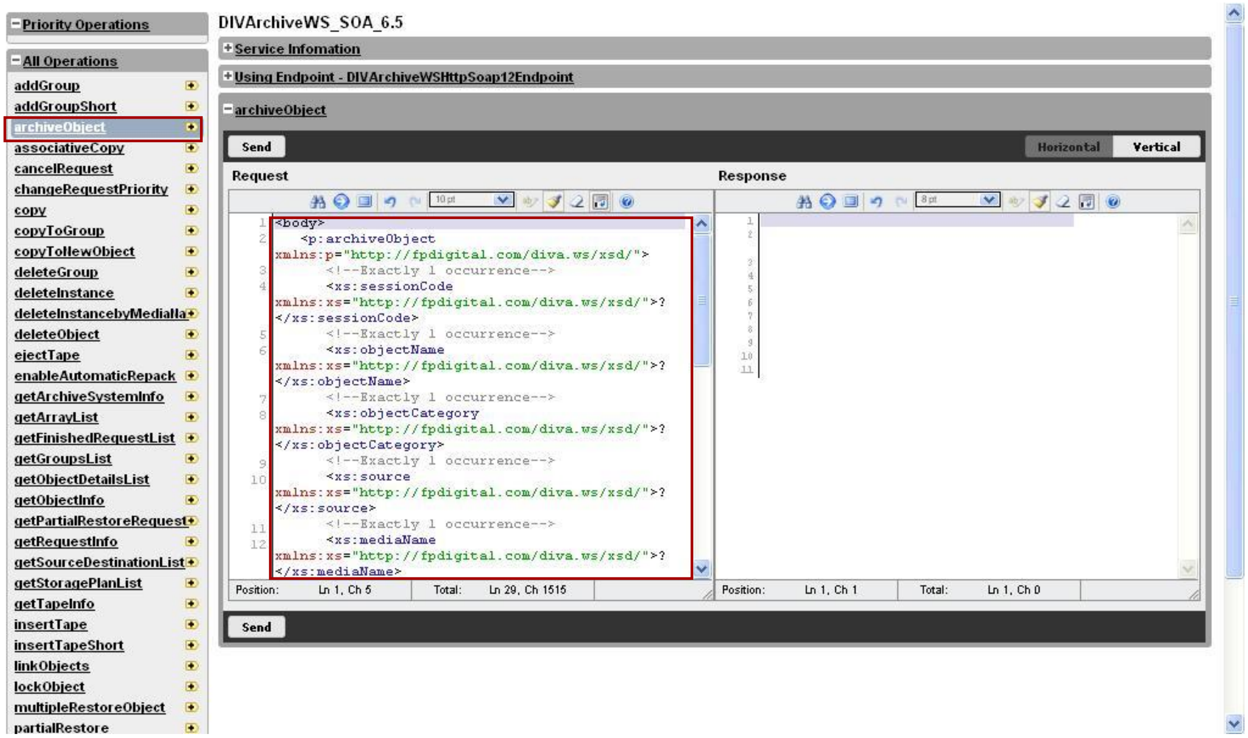
Session Code returned.

5.3.2 Scenario 2: Submitting an `archiveObject` Request

1. Repeat all steps from Scenario 1 (*5.3.1*) by submitting a **`registerClient`** Request to get a valid **`sessionCode`**. All Requests in DIVArchive WS use the **`sessionId`** value as one of the parameters.

2. Similar to Step 1 in Scenario 1, select the **`archiveObject`** Request from the *All Operation List*.

*Figure 16: Select **`archiveObject`** From the All Operations List*



```
<body>
   <p:archiveObject
xmlns:p="http://fpdigital.com/diva.ws/xsd/">
      <!--Exactly 1 occurrence-->
      <xs:sessionCode
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">?
</xs:sessionCode>
      <!--Exactly 1 occurrence-->
      <xs:objectName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">?
</xs:objectName>
      <!--Exactly 1 occurrence-->
      <xs:objectCategory
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">?
</xs:objectCategory>
      <!--Exactly 1 occurrence-->
      <xs:source
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">?
</xs:source>
      <!--Exactly 1 occurrence-->
      <xs:mediaName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">?
</xs:mediaName>
```

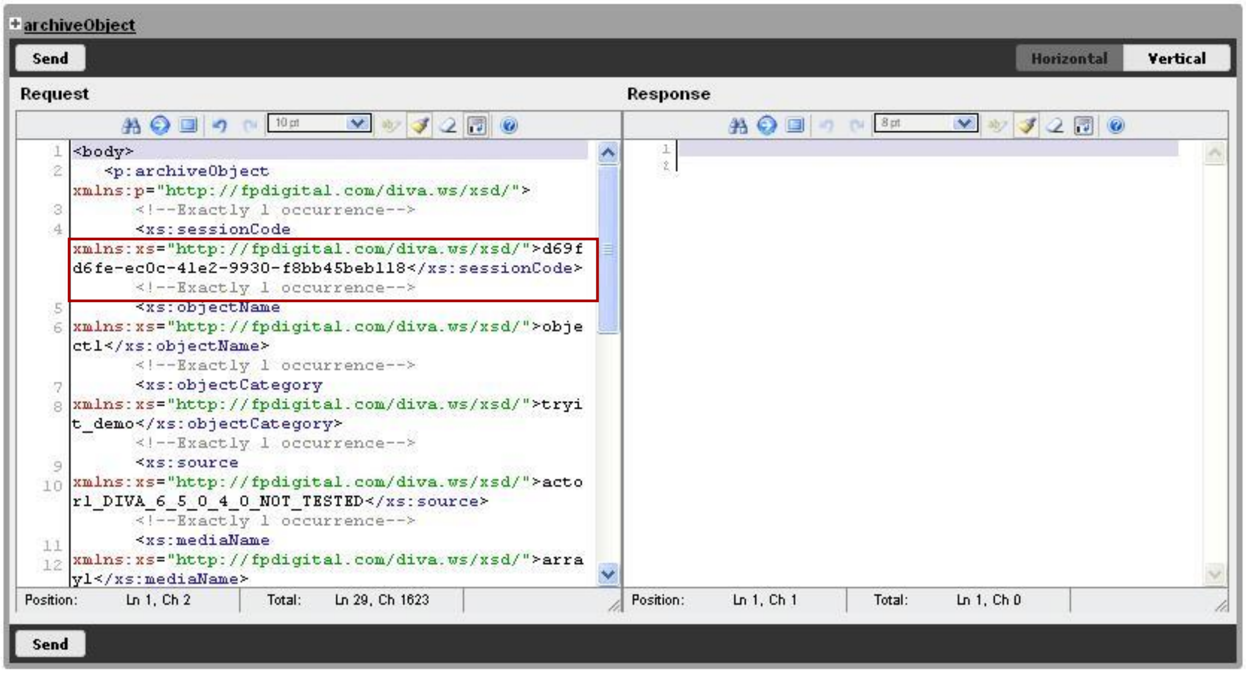
Position: Ln 1, Ch 5 Total: Ln 29, Ch 1515

Position: Ln 1, Ch 1 Total: Ln 1, Ch 0

3. Fill in the *Request Parameters* by replacing "?" symbols with correct values.

   Note: For the `sessionCode` parameter you should provide the value obtained from the `registerClient` Request. It is not necessary to execute the `registerClient` Request again to obtain a new Session Code for each new request being tested. The Session Code, in accordance with the default DIVArchive WS settings, is valid for 30 minutes from last time it was used.

Figure 17: Session Codes are Valid for 30 Minutes by Default

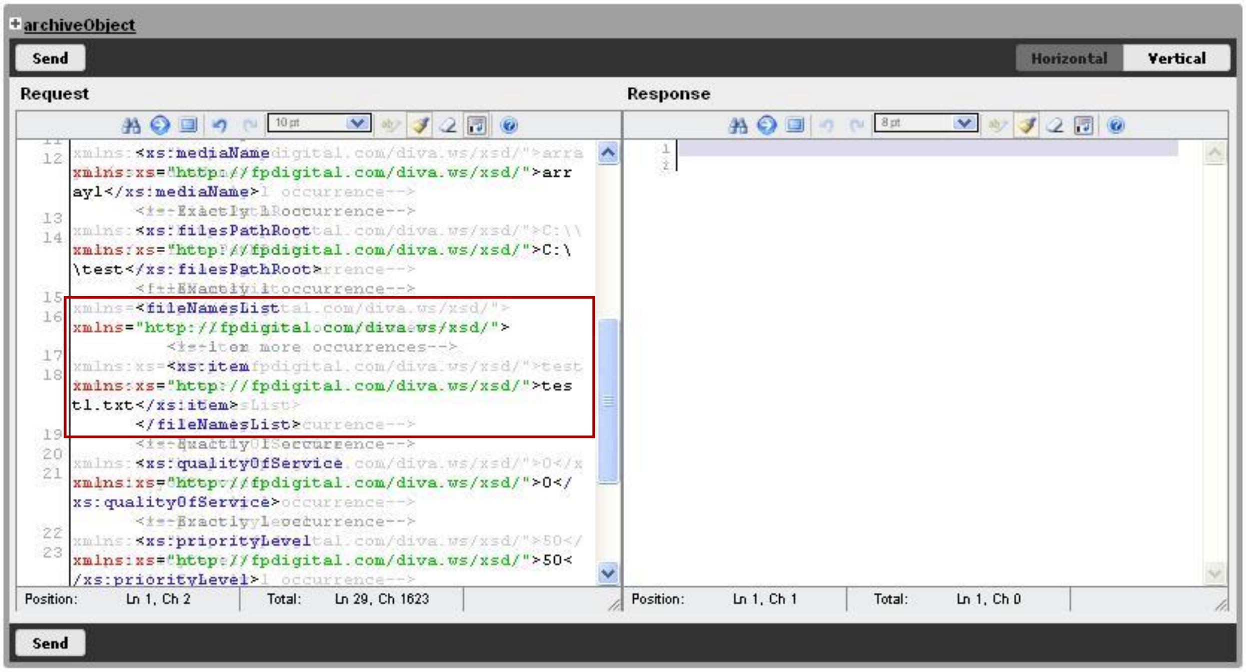


```
<body>
    <p:archiveObject
xmlns:p="http://fpdigital.com/diva.ws/xsd/">
        <!--Exactly 1 occurrence-->
        <xs:sessionCode
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">d69f
d6fe-ec0c-41e2-9930-f8bb45beb118</xs:sessionCode>
        <!--Exactly 1 occurrence-->
        <xs:objectName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">obje
ct1</xs:objectName>
        <!--Exactly 1 occurrence-->
        <xs:objectCategory
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">tryi
t_demo</xs:objectCategory>
        <!--Exactly 1 occurrence-->
        <xs:source
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">acto
r1_DIVA_6_5_0_4_0_NOT_TESTED</xs:source>
        <!--Exactly 1 occurrence-->
        <xs:mediaName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">arra
y1</xs:mediaName>
```

```
Position:   Ln 1, Ch 2    Total:   Ln 29, Ch 1623
Position:   Ln 1, Ch 1    Total:   Ln 1, Ch 0
```

In this request we initially created an *Archive Request* to archive only one file named *test1.txt*. See the figure below.

Figure 18: Archive Request for a Single File

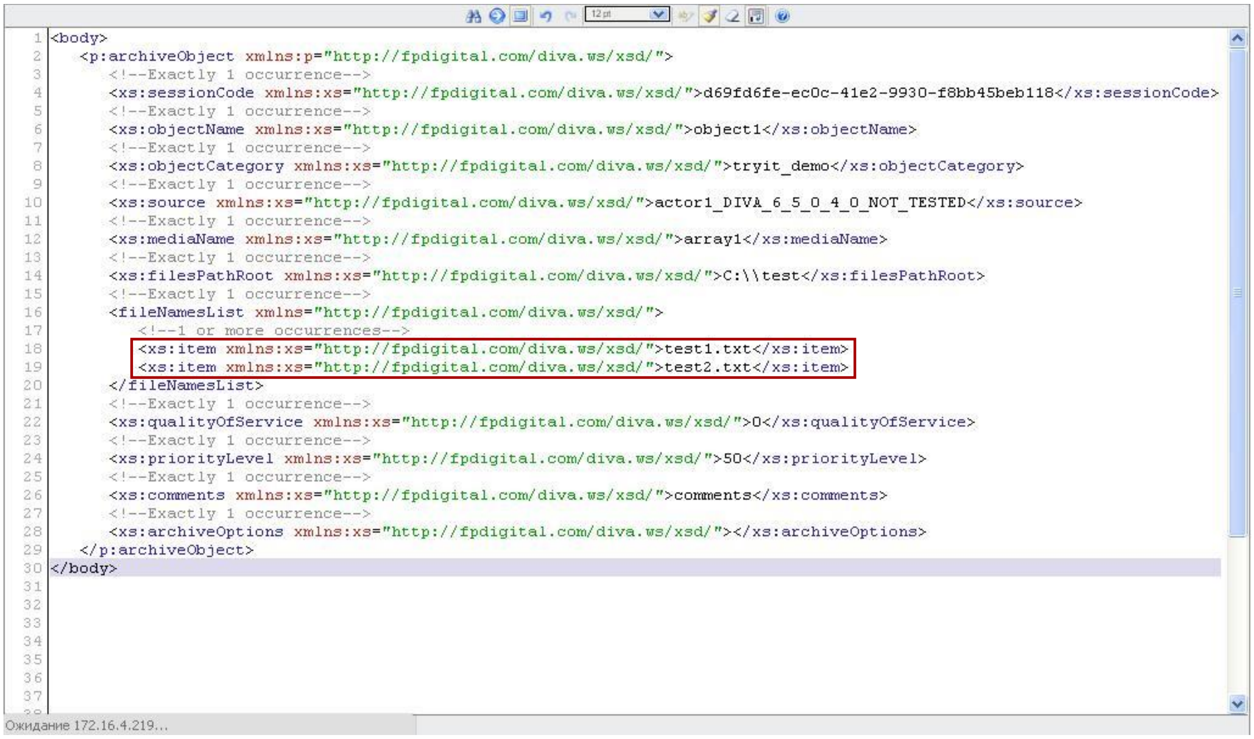


```
        <xs:mediaName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">arr
ay1</xs:mediaName>
        <!--Exactly 1 occurrence-->
        <xs:filesPathRoot
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">C:\
\test</xs:filesPathRoot>
        <!--Exactly 1 occurrence-->
        <fileNamesList
xmlns="http://fpdigital.com/diva.ws/xsd/">
            <!--1 or more occurrences-->
            <xs:item
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">tes
t1.txt</xs:item>
        </fileNamesList>
        <!--Exactly 1 occurrence-->
        <xs:qualityOfService
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">0</
xs:qualityOfService>
        <!--Exactly 1 occurrence-->
        <xs:priorityLevel
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">50<
/xs:priorityLevel>
```

```
Position:   Ln 1, Ch 2    Total:   Ln 29, Ch 1623
Position:   Ln 1, Ch 1    Total:   Ln 1, Ch 0
```

If additional files are to be added, duplicate all of the text inside the `<item>` tag using the additional filenames. The figure below shows the same request, but with two files to archive rather than one (*in the figure below Full Screen Mode was selected in the TryIt GUI*).

Figure 19: Full Screen Mode Showing Multiple File Entries



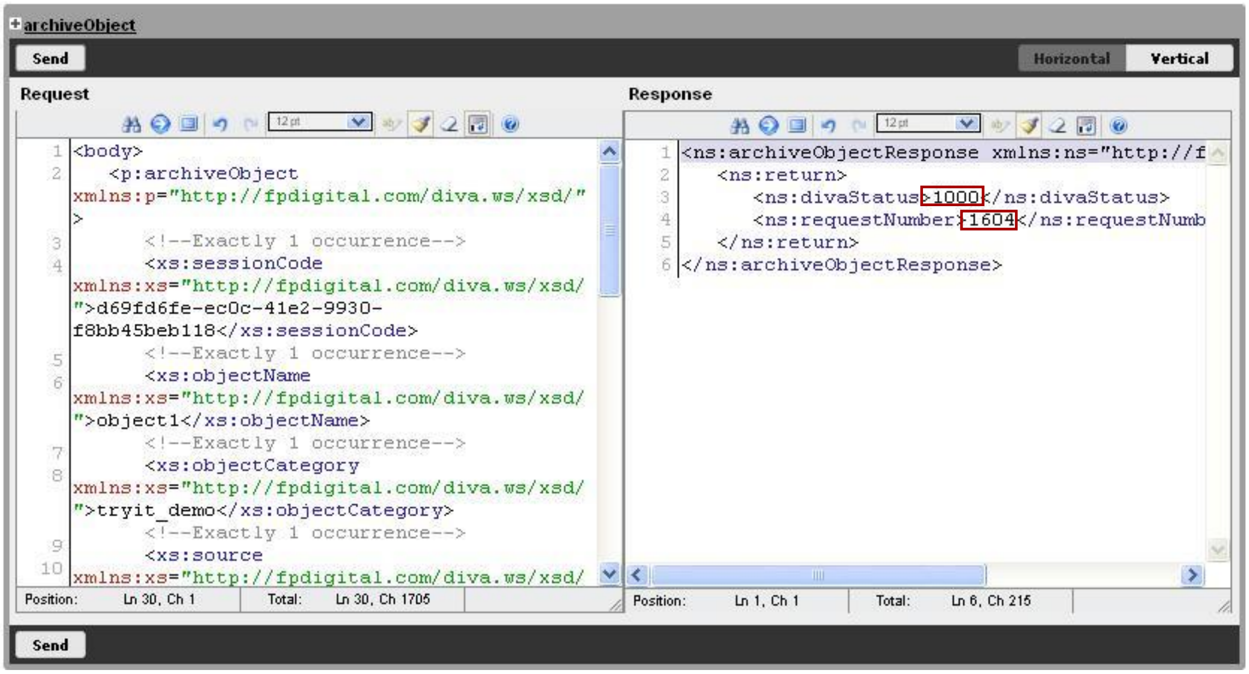
```
<body>
   <p:archiveObject xmlns:p="http://fpdigital.com/diva.ws/xsd/">
      <!--Exactly 1 occurrence-->
      <xs:sessionCode xmlns:xs="http://fpdigital.com/diva.ws/xsd/">d69fd6fe-ec0c-41e2-9930-f8bb45beb118</xs:sessionCode>
      <!--Exactly 1 occurrence-->
      <xs:objectName xmlns:xs="http://fpdigital.com/diva.ws/xsd/">object1</xs:objectName>
      <!--Exactly 1 occurrence-->
      <xs:objectCategory xmlns:xs="http://fpdigital.com/diva.ws/xsd/">tryit_demo</xs:objectCategory>
      <!--Exactly 1 occurrence-->
      <xs:source xmlns:xs="http://fpdigital.com/diva.ws/xsd/">actor1_DIVA_6_5_0_4_0_NOT_TESTED</xs:source>
      <!--Exactly 1 occurrence-->
      <xs:mediaName xmlns:xs="http://fpdigital.com/diva.ws/xsd/">array1</xs:mediaName>
      <!--Exactly 1 occurrence-->
      <xs:filesPathRoot xmlns:xs="http://fpdigital.com/diva.ws/xsd/">C:\\test</xs:filesPathRoot>
      <!--Exactly 1 occurrence-->
      <fileNamesList xmlns="http://fpdigital.com/diva.ws/xsd/">
         <!--1 or more occurrences-->
         <xs:item xmlns:xs="http://fpdigital.com/diva.ws/xsd/">test1.txt</xs:item>
         <xs:item xmlns:xs="http://fpdigital.com/diva.ws/xsd/">test2.txt</xs:item>
      </fileNamesList>
      <!--Exactly 1 occurrence-->
      <xs:qualityOfService xmlns:xs="http://fpdigital.com/diva.ws/xsd/">0</xs:qualityOfService>
      <!--Exactly 1 occurrence-->
      <xs:priorityLevel xmlns:xs="http://fpdigital.com/diva.ws/xsd/">50</xs:priorityLevel>
      <!--Exactly 1 occurrence-->
      <xs:comments xmlns:xs="http://fpdigital.com/diva.ws/xsd/">comments</xs:comments>
      <!--Exactly 1 occurrence-->
      <xs:archiveOptions xmlns:xs="http://fpdigital.com/diva.ws/xsd/"></xs:archiveOptions>
   </p:archiveObject>
</body>
```

Ожидание 172.16.4.219...

4. Submit the request by clicking on the *Send* button. If correct values were provided for the parameters, a response should be received with a **`divaStatus`** value of 1000, which corresponds to the **`DIVA_OK`** status and indicates that the **`archiveRequest`** was submitted successfully into the DIVArchive System.

   The response also contains the Request Number of the submitted request (*in this example its 1604*).

Figure 20: Response Window Showing ***`divaStatus`*** *and* ***`requestNumber`***



```
<body>
   <p:archiveObject
xmlns:p="http://fpdigital.com/diva.ws/xsd/"
>
      <!--Exactly 1 occurrence-->
      <xs:sessionCode
xmlns:xs="http://fpdigital.com/diva.ws/xsd/
">d69fd6fe-ec0c-41e2-9930-
f8bb45beb118</xs:sessionCode>
      <!--Exactly 1 occurrence-->
      <xs:objectName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/
">object1</xs:objectName>
      <!--Exactly 1 occurrence-->
      <xs:objectCategory
xmlns:xs="http://fpdigital.com/diva.ws/xsd/
">tryit_demo</xs:objectCategory>
      <!--Exactly 1 occurrence-->
      <xs:source
xmlns:xs="http://fpdigital.com/diva.ws/xsd/
```

Position: Ln 30, Ch 1 Total: Ln 30, Ch 1705

```
<ns:archiveObjectResponse xmlns:ns="http://f
   <ns:return>
      <ns:divaStatus>1000</ns:divaStatus>
      <ns:requestNumber>1604</ns:requestNumb
   </ns:return>
</ns:archiveObjectResponse>
```

Position: Ln 1, Ch 1 Total: Ln 6, Ch 215

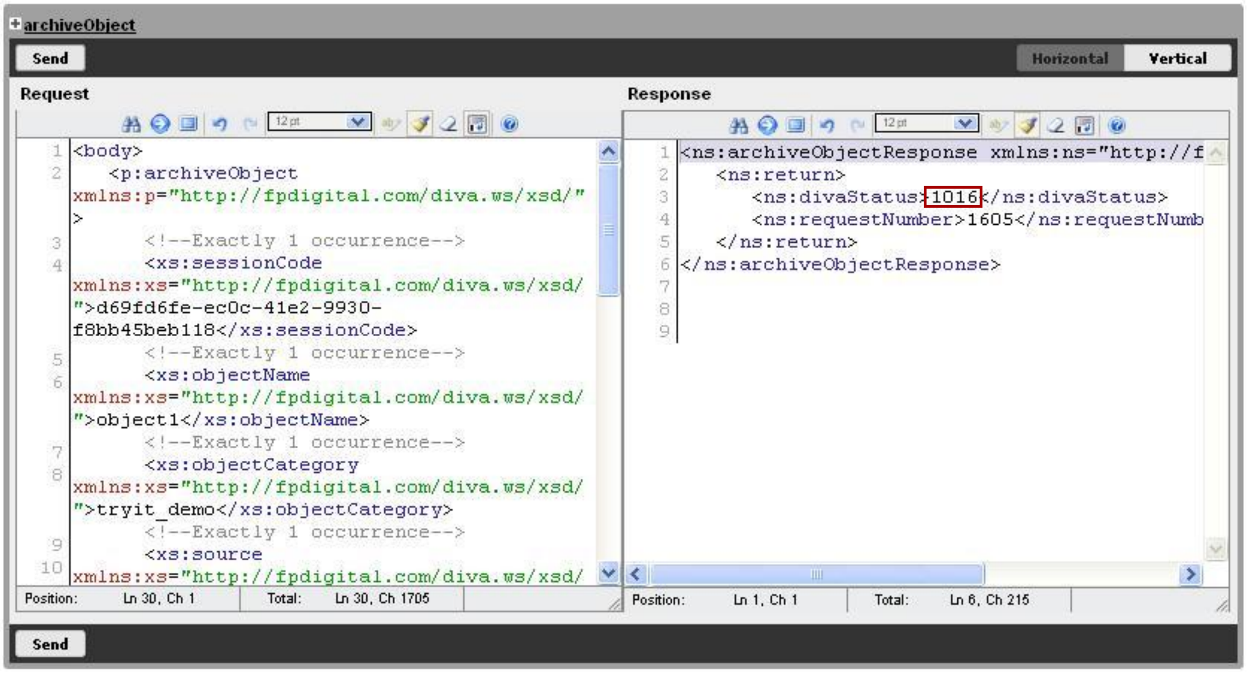
5.3.3 Scenario 3: Errors in the Response

In this scenario some common errors in the response are demonstrated.

5.3.3.1 Sample 1: DIVArchive Errors

In the figure below you can see that the **divaStatus** is not 1000 (***DIVA_OK***) but rather 1016 (***DIVA_ERR_OBJECT_ALREADY_EXISTS***). This response was the result of executing the request from the previous scenario using the same parameters twice in a row. For additional DIVArchive Error Code information please refer to the DIVArchive WS API Reference Manual.

Figure 21: Incorrect ***divaStatus*** *Response Code - Error*



Request:

```
<body>
   <p:archiveObject
xmlns:p="http://fpdigital.com/diva.ws/xsd/"
>
         <!--Exactly 1 occurrence-->
         <xs:sessionCode
xmlns:xs="http://fpdigital.com/diva.ws/xsd/
">d69fd6fe-ec0c-41e2-9930-
f8bb45beb118</xs:sessionCode>
         <!--Exactly 1 occurrence-->
         <xs:objectName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/
">object1</xs:objectName>
         <!--Exactly 1 occurrence-->
         <xs:objectCategory
xmlns:xs="http://fpdigital.com/diva.ws/xsd/
">tryit_demo</xs:objectCategory>
         <!--Exactly 1 occurrence-->
         <xs:source
xmlns:xs="http://fpdigital.com/diva.ws/xsd/
```

Position: Ln 30, Ch 1 Total: Ln 30, Ch 1705

Response:

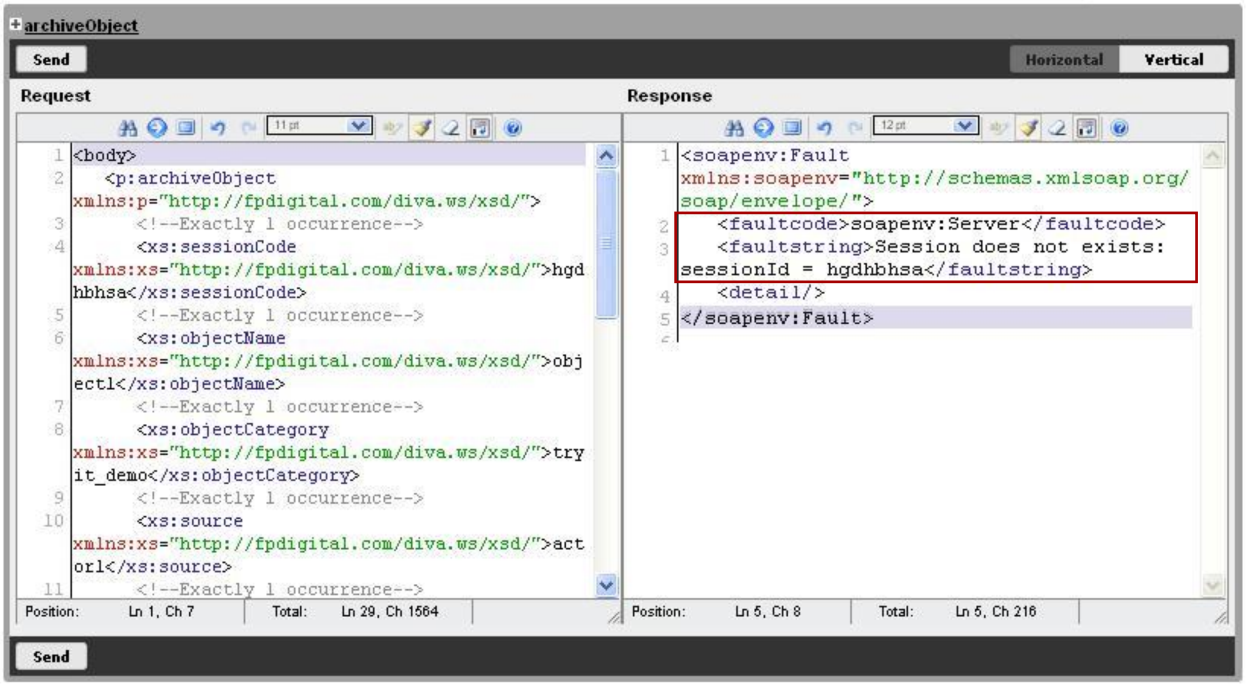
```
<ns:archiveObjectResponse xmlns:ns="http://f
   <ns:return>
      <ns:divaStatus>1016</ns:divaStatus>
      <ns:requestNumber>1605</ns:requestNumb
   </ns:return>
</ns:archiveObjectResponse>
```

Position: Ln 1, Ch 1 Total: Ln 6, Ch 215

5.3.3.2 Sample 2: Invalid Session Code

If you provide an invalid or expired **Session Code** you'll receive the error message shown in the figure below.

Figure 22: Invalid Session Code - Error



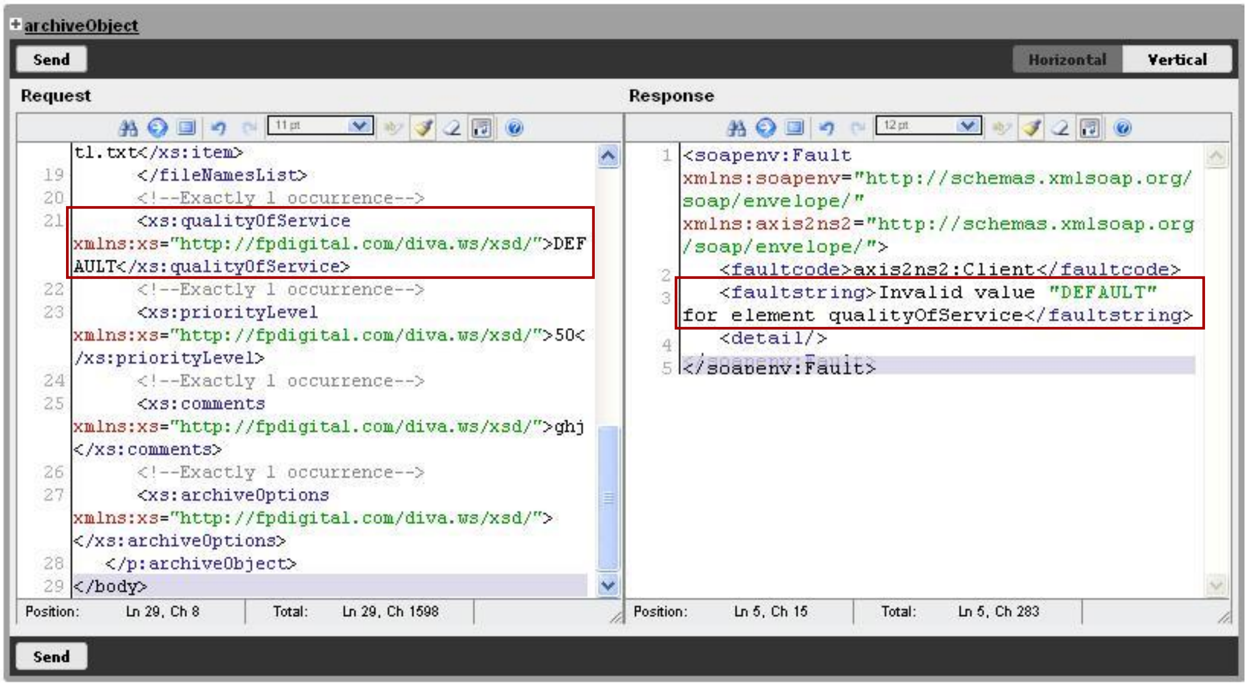
```
<body>
   <p:archiveObject
xmlns:p="http://fpdigital.com/diva.ws/xsd/">
      <!--Exactly 1 occurrence-->
      <xs:sessionCode
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">hgd
hbhsa</xs:sessionCode>
      <!--Exactly 1 occurrence-->
      <xs:objectName
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">obj
ect1</xs:objectName>
      <!--Exactly 1 occurrence-->
      <xs:objectCategory
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">try
it_demo</xs:objectCategory>
      <!--Exactly 1 occurrence-->
      <xs:source
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">act
or1</xs:source>
      <!--Exactly 1 occurrence-->
```

```
<soapenv:Fault
xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/">
   <faultcode>soapenv:Server</faultcode>
   <faultstring>Session does not exists:
sessionId = hgdhbhsa</faultstring>
   <detail/>
</soapenv:Fault>
```

5.3.3.3 Sample 3: Invalid Parameter Type

When an invalid value for a parameter in the request is provided, an error will be received similar to the one shown on the figure below. In this example a correct **`archiveObject`** request was submitted, but it used the string value **`DEFAULT`** for the **`qualityOfService`** parameter while (*according to DIVArchive specifications*) only integer values should be used for this parameter.

Figure 23: Incorrect/Invalid Parameter Value Entered - Error



Request:

```
tl.txt</xs:item>
        </fileNamesList>
        <!--Exactly 1 occurrence-->
        <xs:qualityOfService
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">DEF
AULT</xs:qualityOfService>
        <!--Exactly 1 occurrence-->
        <xs:priorityLevel
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">50<
/xs:priorityLevel>
        <!--Exactly 1 occurrence-->
        <xs:comments
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">ghj
</xs:comments>
        <!--Exactly 1 occurrence-->
        <xs:archiveOptions
xmlns:xs="http://fpdigital.com/diva.ws/xsd/">
</xs:archiveOptions>
    </p:archiveObject>
</body>
```

Position: Ln 29, Ch 8 Total: Ln 29, Ch 1598

Response:

```
<soapenv:Fault
xmlns:soapenv="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:axis2ns2="http://schemas.xmlsoap.org
/soap/envelope/">
   <faultcode>axis2ns2:Client</faultcode>
   <faultstring>Invalid value "DEFAULT"
for element qualityOfService</faultstring>
   <detail/>
</soapenv:Fault>
```

Position: Ln 5, Ch 15 Total: Ln 5, Ch 283

5.4 Oracle WS Client Applications

5.4.1 Command Line Tool for `getObjectDetailsList` Requests

A new command line tool has been added to the DIVArchive WS API and is included in the DIVAS package. The tool allows the user to send requests for the **`getObjectDetailsList`** API call to DIVArchive.

Notes:

- **The tool is ONLY applicable to this request.**
- **The `JAVA_HOME` variable must be set in order to run the tool.**

The program can be operated in either **Interactive Mode** or **Automatic Mode**. **Interactive Mode** prompts the user for input while the **Automatic Mode** does not and each mode has its own set of available options/parameters.

To execute the command line interface:

1. Navigate to **`<DIVAS_HOME>\tools\DIVArchiveWSClient.<version>\bin`**.
2. Execute: **`run.bat <options>`**

Interactive Mode is active by default even if no parameters are specified. The default connection is **`localhost:9763`**; however connection parameters may be identified using the command:

```
run <host:port>
```

Example: **`run 192.168.1.1:9763`**

Automatic Mode can be initialized using the **`–a`** option. Some parameters are required while others are optional. To execute the tool in **Automatic Mode** use the command:

```
run –a {-s|-d} [<host:port> <maxListSize> <waitInterval>
<objectCategory> <objectName>]
```

The parameters above that are enclosed in **`{}`** are required parameters and the parameters enclosed in **`[]`** are optional parameters.

The table below describes the different **`<options>`** available when using the command line interface tool in **Automatic Mode**, and whether or not they are required or optional.

Table 7: `getObjectDetailsList` Command Line Tool Options

Parameter	Description	Required or Optional
`-a`	Initiates Automatic Mode.	Required to run in Automatic Mode.
`-s`	Type of output details: prints the sum of the number of objects only.	Required
`-d`	Type of output details: prints object details along with the sum of the number of objects.	Required

Parameter	Description	Required or Optional
`host:port`	The host IP Address and port number to use for the connection. The default is localhost:9763.	Optional
`maxListSize`	The maximum size of the list being reported. The default is 5.	Optional
`waitInterval`	The maximum time (*in milliseconds – ms*) to wait for the report to be generated. The default is 1000ms.	Optional
`objectCategory`	The Object Category to report. The default is * (*all categories*).	Optional
`objectName`	The Object Name to report. The default is * (*all categories*).	Optional

Example command lines:

- **`run -a -s`**
  - Runs the tool in Automatic Mode and prints the sum of the number of objects.
- **`run -a -d 192.168.1.15:9763`**
  - Runs the tool in Automatic Mode and prints the object details along with the sum of the number of objects and connects to 192.168.1.15 using port 9763.
- **`run -a -s 192.168.1.15:9763 200`**
  - Runs the tool in Automatic Mode, prints the sum of the number of objects, connects to 192.168.1.15 using port 9763, and the maximum list size returned is 200.
- **`run -a -s 192.168.1.15:9763 200 100`**
  - Runs the tool in Automatic Mode, prints the object details along with the sum of the number of objects, connects to 192.168.1.15 using port 9763, the maximum list size returned is 200 and the wait interval is 100ms.
- **`run -a -s 192.168.1.15:9763 200 100 testCategory`**
  - Runs the tool in Automatic Mode, prints the object details along with the sum of the number of objects, connects to 192.168.1.15 using port 9763, the maximum list size returned is 200, the wait interval is 100ms, and the Object Category being reported on is **`testCategory`**.
- **`run -a -s 192.168.1.15:9763 200 100 testCategory testObject`**
  - Runs the tool in Automatic Mode, prints the object details along with the sum of the number of objects, connects to 192.168.1.15 using port 9763, the maximum list size returned is 200, the wait interval is 100ms, the Object Category being reported on is **`testCategory`**, and the object being reported on is named **`testObject`**.

5.5 Third Party WS Client Applications

There are three applications that are recommended for use as a DIVArchive API Client:

- soapUI
- soapUI Pro
- WcfTestClient

The table below describes these applications:

Table 8: Recommended DIVArchive WS API Client Applications

Feature	soapUI	soapUI Pro	WcfTestClient	Description
Reads list of all requests from DIVArchive WS WSDL.	Yes	Yes	Yes	The user provides the URL to the WSDL and the application creates a list of all DIVArchive WS API requests with their properties.
Sends request and views the response	Yes	Yes	Yes	For all requests available in DIVArchive WS API it is possible to send requests and to view the response.
Inputs requests parameters in XML format.	Yes	Yes	Yes	User can edit the default XML request and provide required values for its parameters.
Inputs request parameters in a GUI.	No	Yes	Yes	Instead of editing XML, the user can use more a convenient GUI to enter requests parameters and review response values.
Displays additional information about the request parameters (*parameter type, nullability, etc.*)	No	Yes	Yes	The response to the request includes additional information related to the request's parameters.
Free version availability.	Yes	Free 14-day trial.	Both free and commercial options exist.	Availability of free vs. commercial versions of the application.

5.5.1 Obtaining the Tools

Both the **soapUI** and **soapUI Pro** can be downloaded from http://www.soapui.org/.

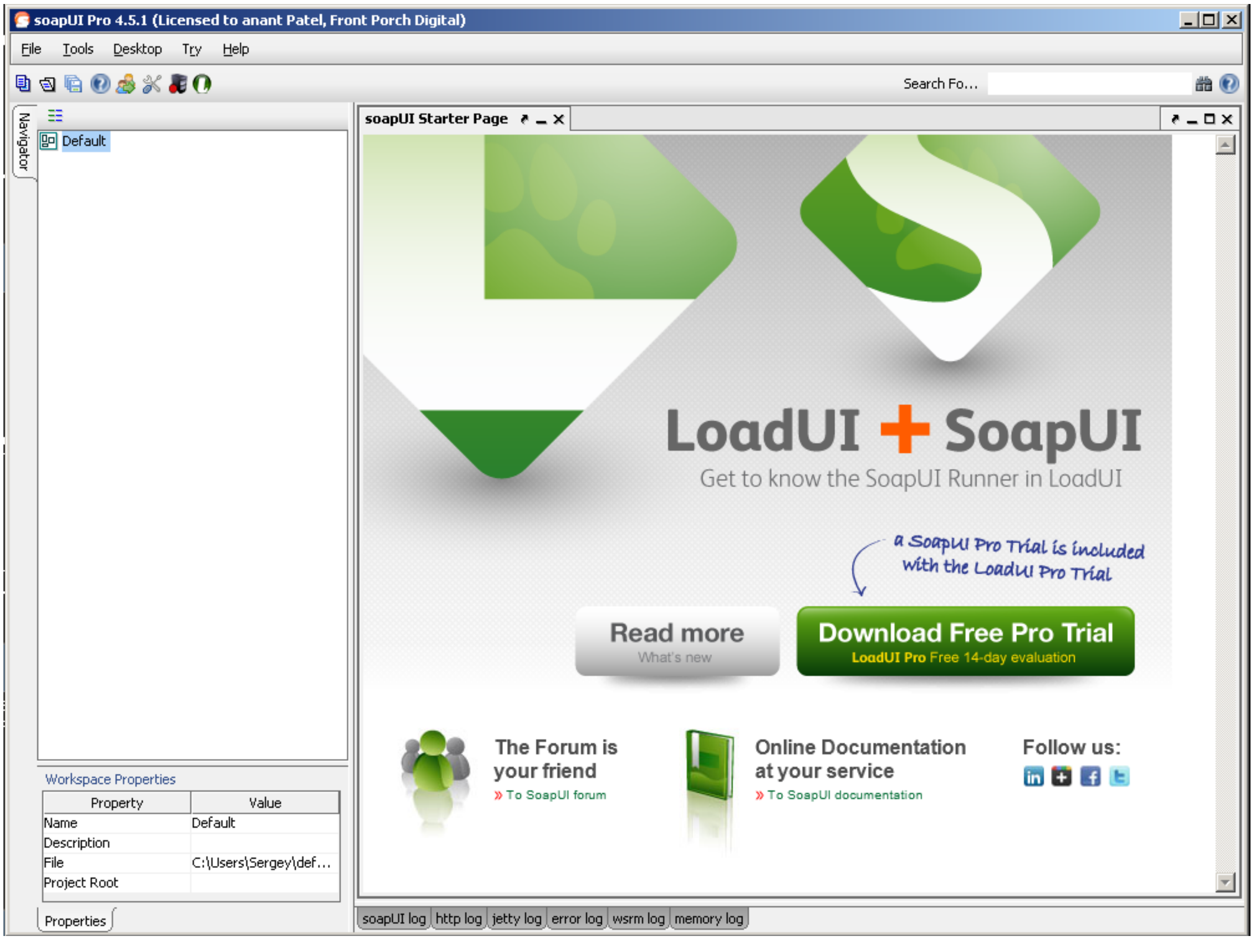
The **WcfTestClient Tool** is part of the **MS Visual Studio .Net** package. The latest free version can be found in **MS Visual Studio Express .Net 2008** and can be downloaded from http://go.microsoft.com/?linkid=7729278. Another option is to use the newest commercial version of **MS Visual Studio**. After **MS Visual Studio** is installed, locate **`WcfTestClient.exe`** file and execute it to run the application.

5.5.2 Using the Tools to Run Requests

5.5.2.1 soapUI and soapUI Pro

1. Start **soapUI** / **soapUI Pro** using the Desktop icon created during installation.

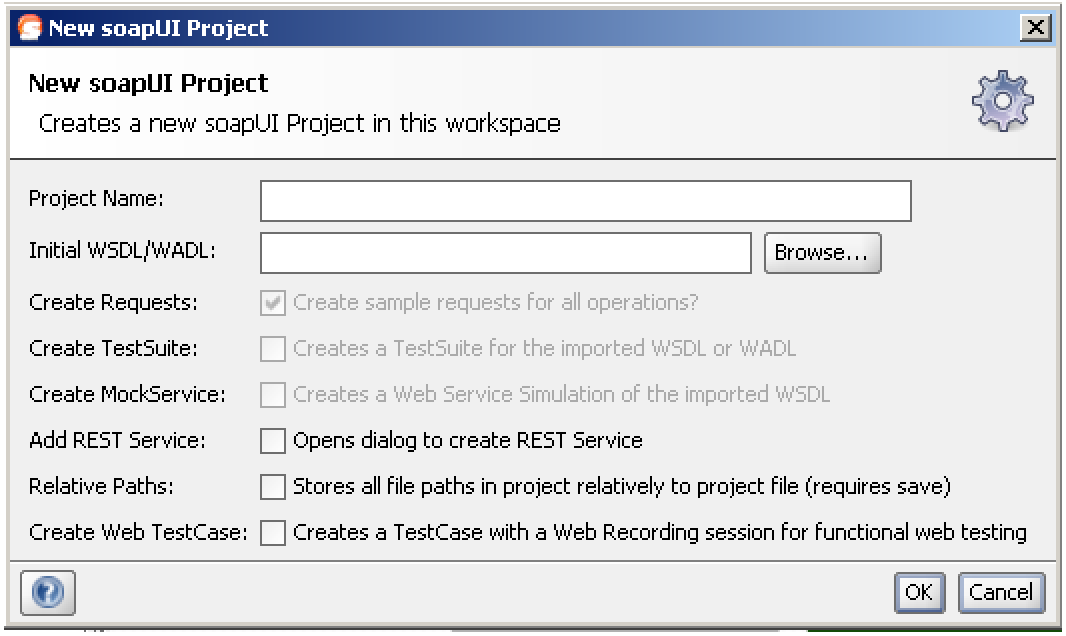
Figure 24: soapUI / soapUI Pro Opening Screen



Property	Value
Name	Default
Description	
File	C:\Users\Sergey\def...
Project Root	

2. From the Menu Bar, select **File** then **New soapUI Project**
    a. The dialog window shown below will appear.

Figure 25: New soapUI Project Window


New soapUI Project

New soapUI Project

Creates a new soapUI Project in this workspace

Project Name:

Initial WSDL/WADL:

Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: □ Creates a TestSuite for the imported WSDL or WADL

Create MockService: □ Creates a Web Service Simulation of the imported WSDL

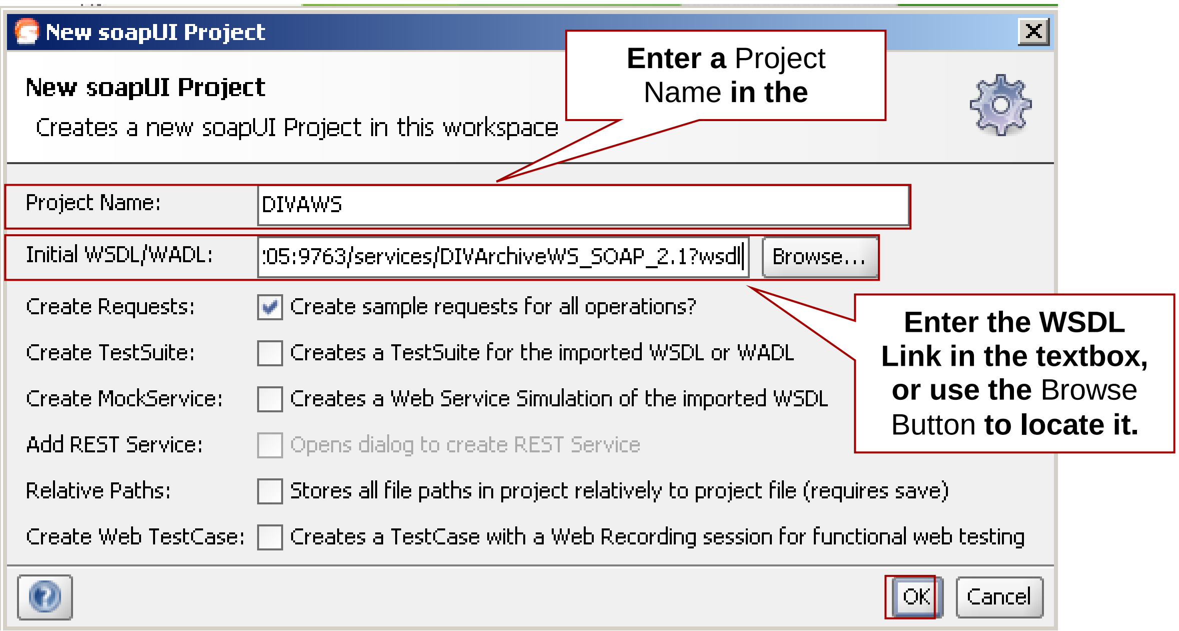
Add REST Service: □ Opens dialog to create REST Service

Relative Paths: □ Stores all file paths in project relatively to project file (requires save)

Create Web TestCase: □ Creates a TestCase with a Web Recording session for functional web testing

3. Enter a **Project Name** (*any name is acceptable*) in the textbox, and provide the link to the WSDL for DIVArchive WS.

Figure 26: Enter the Project Name and WSDL Link


New soapUI Project

New soapUI Project

Creates a new soapUI Project in this workspace

Project Name: DIVAWS

Initial WSDL/WADL: :05:9763/services/DIVArchiveWS_SOAP_2.1?wsdl

Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: □ Creates a TestSuite for the imported WSDL or WADL

Create MockService: □ Creates a Web Service Simulation of the imported WSDL

Add REST Service: □ Opens dialog to create REST Service

Relative Paths: □ Stores all file paths in project relatively to project file (requires save)

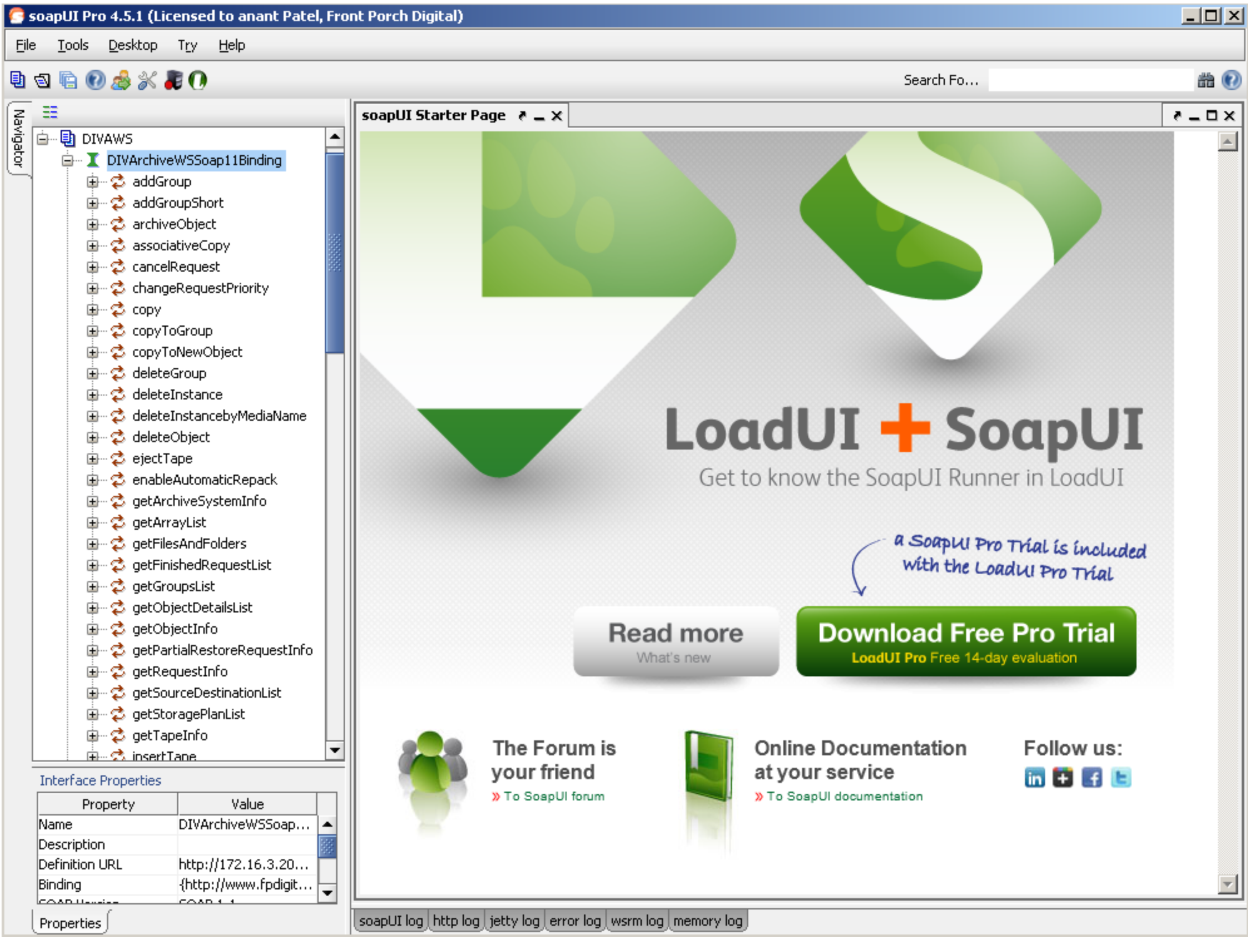
Create Web TestCase: □ Creates a TestCase with a Web Recording session for functional web testing

Enter a Project Name **in the**

Enter the WSDL Link in the textbox, or use the Browse Button **to locate it.**

4. Click the **OK Button**. The application loads information from the WSDL and generate a list of all requests available as shown below.

Figure 27: All Available Requests Screen

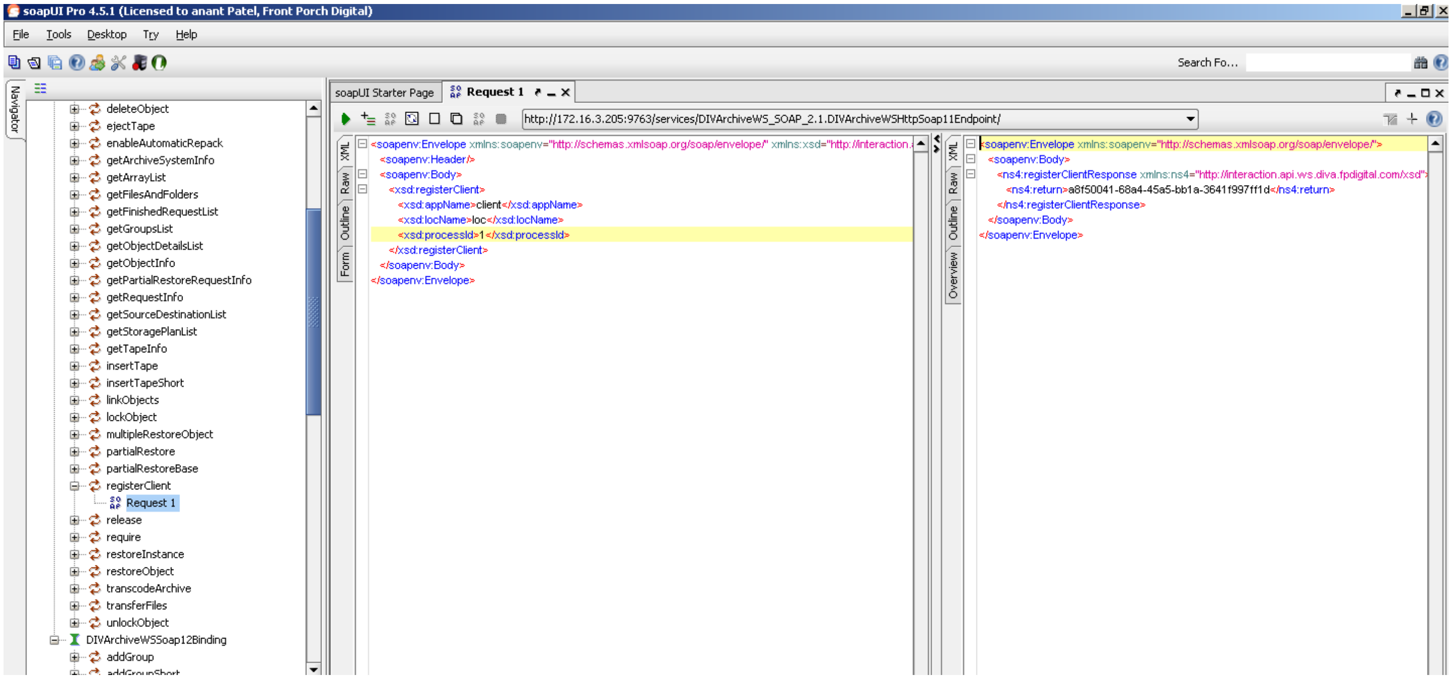


- DIVAWS
  - DIVArchiveWSSoap11Binding
    - addGroup
    - addGroupShort
    - archiveObject
    - associativeCopy
    - cancelRequest
    - changeRequestPriority
    - copy
    - copyToGroup
    - copyToNewObject
    - deleteGroup
    - deleteInstance
    - deleteInstancebyMediaName
    - deleteObject
    - ejectTape
    - enableAutomaticRepack
    - getArchiveSystemInfo
    - getArrayList
    - getFilesAndFolders
    - getFinishedRequestList
    - getGroupsList
    - getObjectDetailsList
    - getObjectInfo
    - getPartialRestoreRequestInfo
    - getRequestInfo
    - getSourceDestinationList
    - getStoragePlanList
    - getTapeInfo
    - insertTape

Property	Value
Name	DIVArchiveWSSoap...
Description	
Definition URL	http://172.16.3.20...
Binding	{http://www.fpdigit...
SOAP Version	SOAP 1.1

5. Select any request from the list in left panel, double-click it and the request's parameters will be displayed; enter the correct values and send the request.

Figure 28: Running a Request



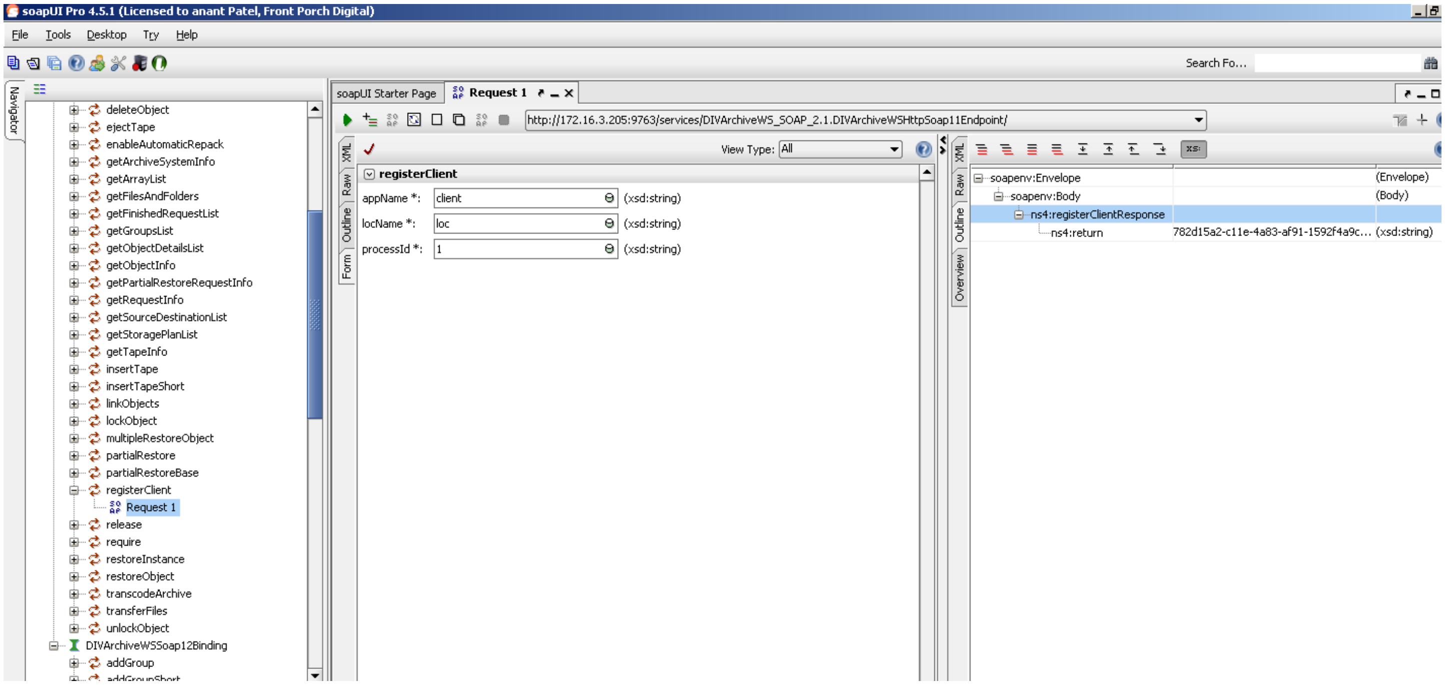
http://172.16.3.205:9763/services/DIVArchiveWS_SOAP_2.1.DIVArchiveWSHttpSoap11Endpoint/

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://interaction.
  <soapenv:Header/>
  <soapenv:Body>
    <xsd:registerClient>
      <xsd:appName>client</xsd:appName>
      <xsd:locName>loc</xsd:locName>
      <xsd:processId>1</xsd:processId>
    </xsd:registerClient>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns4:registerClientResponse xmlns:ns4="http://interaction.api.ws.diva.fpdigital.com/xsd">
      <ns4:return>a8f50041-68a4-45a5-bb1a-3641f997ff1d</ns4:return>
    </ns4:registerClientResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

The **soapUI Pro** application offers a **Form Editor** for requests (*shown below*), which is more convenient than working with XML directly:

Figure 29: soapUI Pro Form Editor Screen



```
deleteObject
ejectTape
enableAutomaticRepack
getArchiveSystemInfo
getArrayList
getFilesAndFolders
getFinishedRequestList
getGroupsList
getObjectDetailsList
getObjectInfo
getPartialRestoreRequestInfo
getRequestInfo
getSourceDestinationList
getStoragePlanList
getTapeInfo
insertTape
insertTapeShort
linkObjects
lockObject
multipleRestoreObject
partialRestore
partialRestoreBase
registerClient
    Request 1
release
require
restoreInstance
restoreObject
transcodeArchive
transferFiles
unlockObject
DIVArchiveWSSoap12Binding
    addGroup
    addGroupShort
```

http://172.16.3.205:9763/services/DIVArchiveWS_SOAP_2.1.DIVArchiveWSHttpSoap11Endpoint/

View Type: All

registerClient

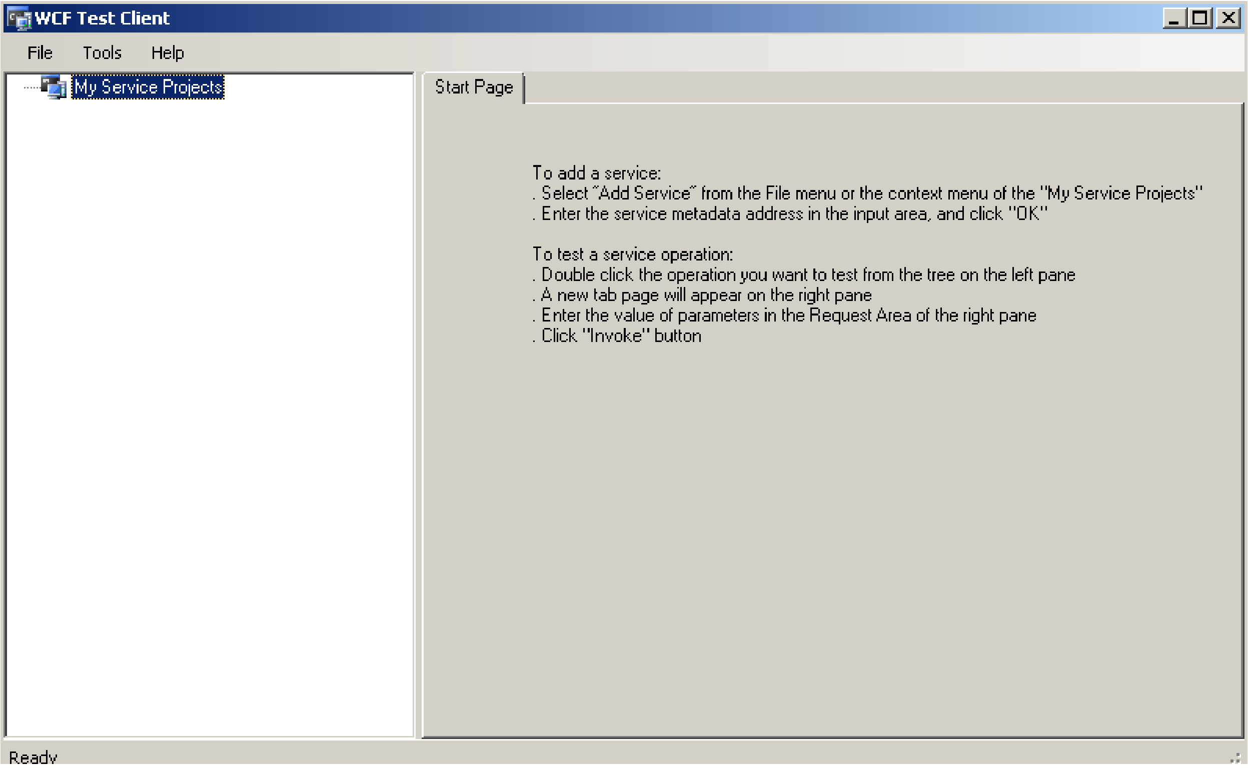
appName *:	client	(xsd:string)
locName *:	loc	(xsd:string)
processId *:	1	(xsd:string)

soapenv:Envelope		(Envelope)
soapenv:Body		(Body)
ns4:registerClientResponse		
ns4:return	782d15a2-c11e-4a83-af91-1592f4a9c...	(xsd:string)

5.5.2.2 WcfTestClient

1. Locate the `WcfTestClient.exe` file in the **MS Visual Studio** installation and execute the program.

Figure 30: WcfTestClient Opening Screen

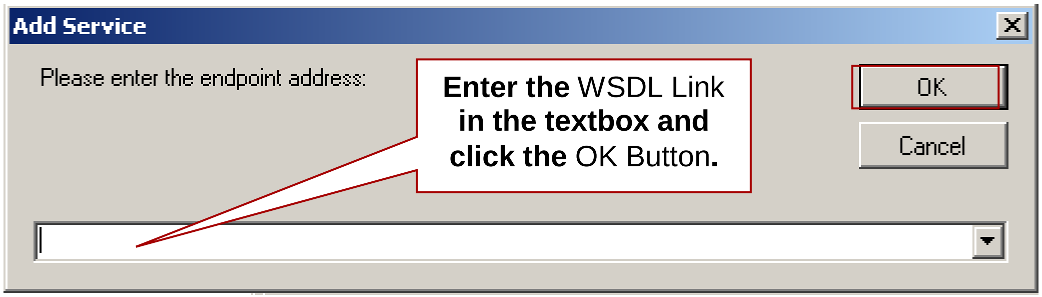


To add a service:
. Select "Add Service" from the File menu or the context menu of the "My Service Projects"
. Enter the service metadata address in the input area, and click "OK"

To test a service operation:
. Double click the operation you want to test from the tree on the left pane
. A new tab page will appear on the right pane
. Enter the value of parameters in the Request Area of the right pane
. Click "Invoke" button

2. From the **Menu Bar**, select **File** then **Add Service**.
   a. The dialog window shown below will appear.
   b. Provide the **WSDL Link** in the textbox and press the **OK Button**.

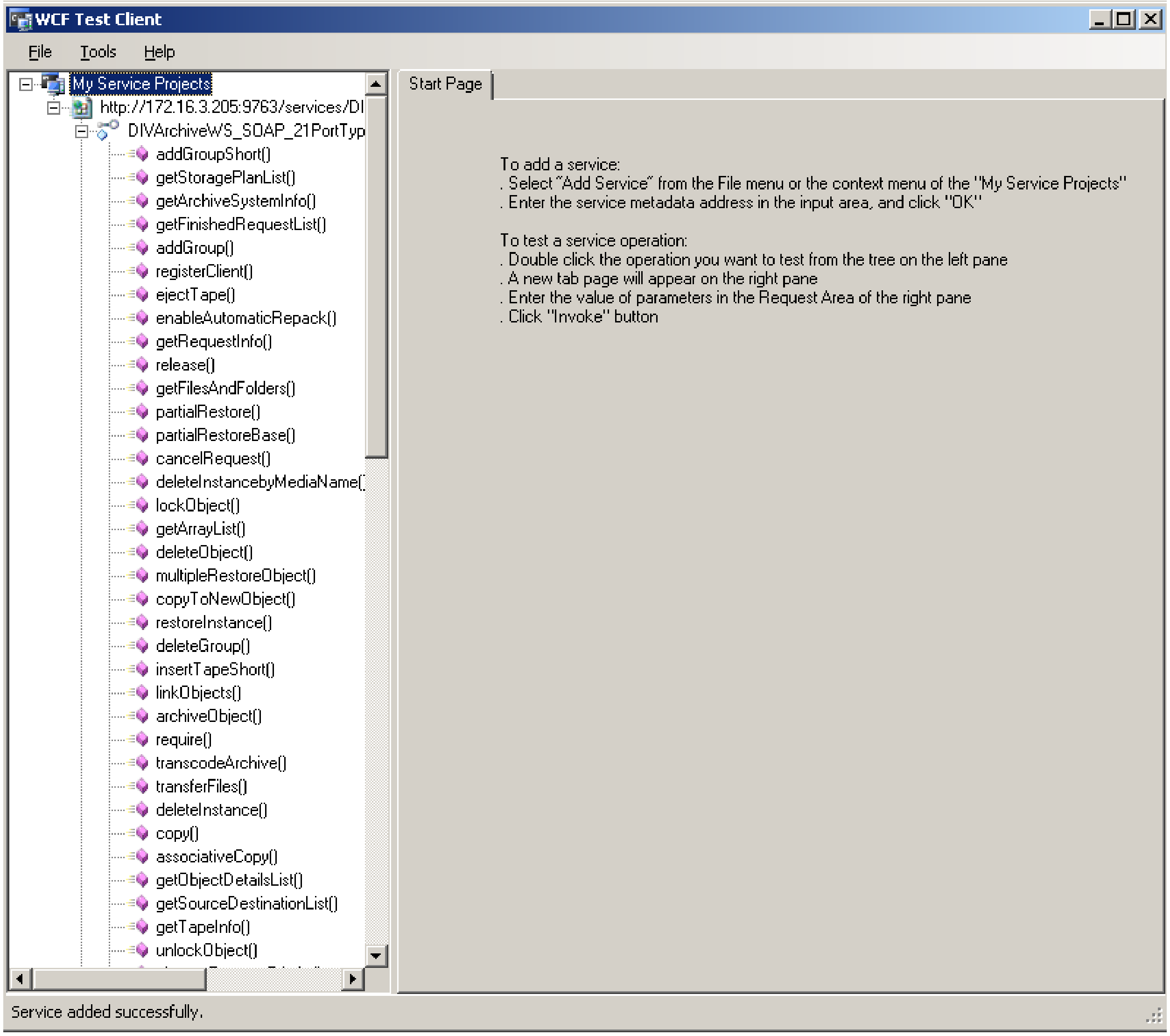
Figure 31: Add Service Window



Please enter the endpoint address:

Enter the WSDL Link **in the textbox and click the** OK Button**.**

Figure 32: All Available Requests Screen



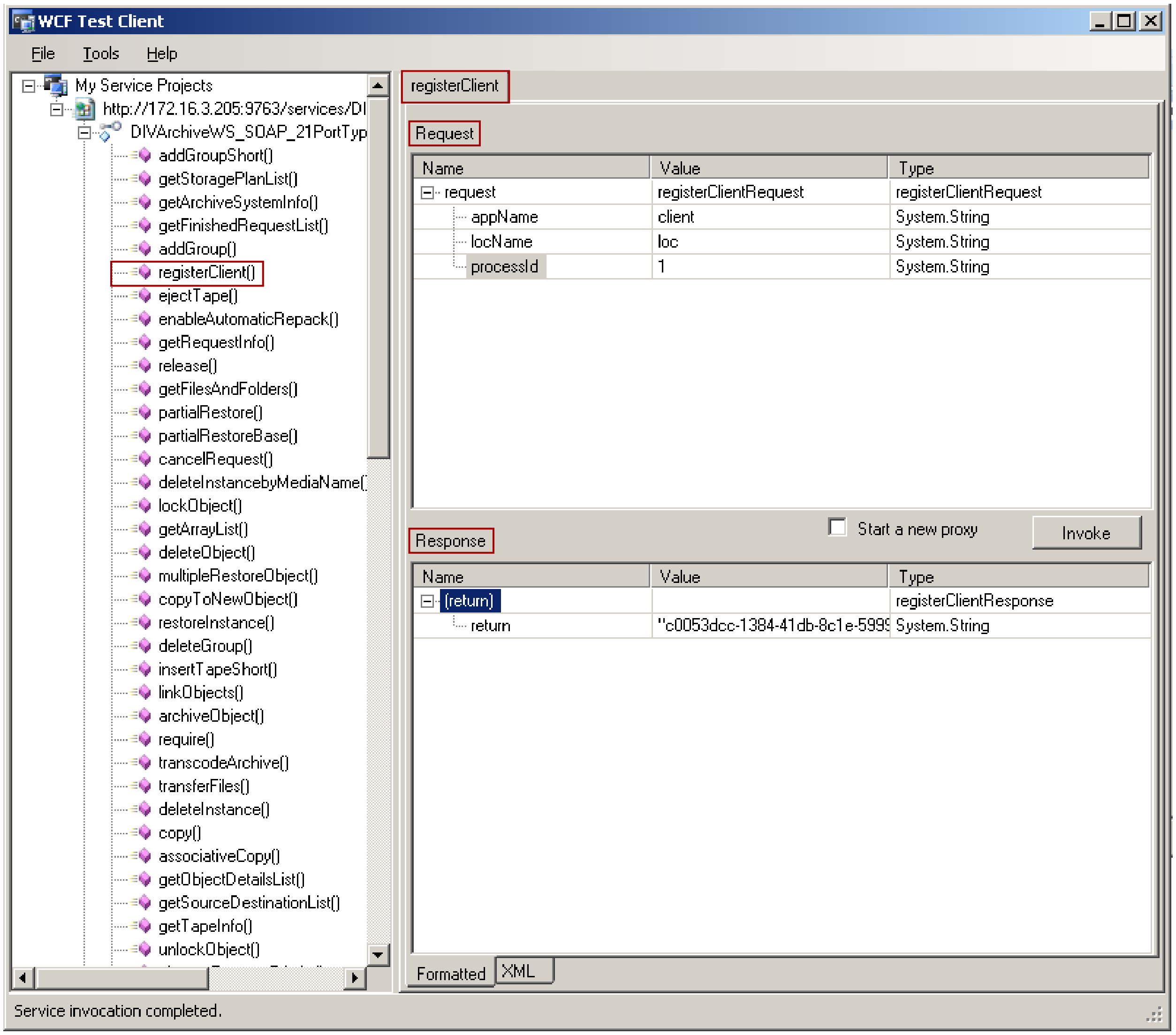
To add a service:
. Select "Add Service" from the File menu or the context menu of the "My Service Projects"
. Enter the service metadata address in the input area, and click "OK"

To test a service operation:
. Double click the operation you want to test from the tree on the left pane
. A new tab page will appear on the right pane
. Enter the value of parameters in the Request Area of the right pane
. Click "Invoke" button

Service added successfully.

3. Select the desired request for testing from the left panel, provide the correct parameters and run the request.

Figure 33: Request and Response Screen



My Service Projects
http://172.16.3.205:9763/services/DI
DIVArchiveWS_SOAP_21PortTyp
addGroupShort()
getStoragePlanList()
getArchiveSystemInfo()
getFinishedRequestList()
addGroup()
registerClient()
ejectTape()
enableAutomaticRepack()
getRequestInfo()
release()
getFilesAndFolders()
partialRestore()
partialRestoreBase()
cancelRequest()
deleteInstancebyMediaName(
lockObject()
getArrayList()
deleteObject()
multipleRestoreObject()
copyToNewObject()
restoreInstance()
deleteGroup()
insertTapeShort()
linkObjects()
archiveObject()
require()
transcodeArchive()
transferFiles()
deleteInstance()
copy()
associativeCopy()
getObjectDetailsList()
getSourceDestinationList()
getTapeInfo()
unlockObject()

Request

Name	Value	Type
request	registerClientRequest	registerClientRequest
appName	client	System.String
locName	loc	System.String
processId	1	System.String

☐ Start a new proxy

Response

Name	Value	Type
(return)		registerClientResponse
return	"c0053dcc-1384-41db-8c1e-599	System.String

Service invocation completed.

6 Frequently Asked Questions

6.1 How do I uninstall the DIVArchive WS API?

1. From the command line run:

   ```
   %DIVAS_HOME%\bin\divas.bat uninstall [service-name]
   ```

2. Delete the folder containing the installed copy of DIVAS.

6.2 What is the process for upgrading DIVArchive WS?

Upgrading the DIVArchive WS installation uses the same procedure as the standard DIVArchive upgrade process. Please refer to 0 for upgrade instructions.

6.3 How do I configure DIVArchive WS to allow access from the VPN subnetwork with different IP settings?

1. In DIVArchive WS configuration define the **`HostName`** parameter as **`ws.host`** or as any other name (*please see 3.2 for more details*).
2. In both the VPN subnetwork and the main network add **`ws.host`** to the DNS configuration using the correct IP settings.
3. When the client connects to DIVArchive WS, it will receive the WSDL with the correct endpoint - **`ws.host.`**

6.4 Why am I receiving errors when using the TryIt Tool with SOAP?

Answer:

Currently the WSO2 TryIt Tool is not 100% compatible with SOAP. This is due to issues with their code and has nothing to do with DIVArchive. Please use the SOAPUI Tool for testing SOAP interactions.

APPENDIX

A1 Default Configuration File

This is a sample of a default configuration file:

```
# DIVArchive instance host.
# Example: 127.0.0.1
DIVARCHIVE_HOST=127.0.0.1

# DIVArchive instance port.
# Example: 9000
DIVARCHIVE_PORT=9000

# DIVArchive instance maximum simultaneous requests.
# Example: 300
DIVARCHIVE_MAX_SIMULTANEOUS_REQUESTS=300

# DIVArchive instance response timeout in minutes.
# Example: 10
DIVARCHIVE_RESPONSE_TIMEOUT_IN_MINUTES=10

# DIVArchive instance login.
# Example: divalogin
DIVARCHIVE_USER_NAME=diva

# DIVArchive instance password in plain decrypted form.
# Example: divapassword
DIVARCHIVE_PASSWORD=diva

# Application name used for connection with DIVArchive system.
# Example: DIVArchiveWS_1.0
#DIVARCHIVE_APPLICATTION_NAME=${DIVARCHIVE_APPLICATTION_NAME}_1.0

# Time interval in minutes. It shows how often DIVArchive session
should be checked and recreate in case of disconnection.
# Example: 1
DIVARCHIVE_SESSION_WATCHER_INTERVAL_IN_MINUTES=1

# Time interval in minutes. Session expires if it does not used more
than given interval.
# Example: 30
SESSION_STORAGE_TIMEOUT_IN_MINUTES=30

# Time interval in minutes. It shows how often session sweeper
should dispose expired sessions.
# Example: 5
SESSION_STORAGE_SWEEPER_INTERVAL_IN_MINUTES=5

# Time interval in minutes. It shows how often monitoring service
writes data to log file.
# Example: 1
MONITORING_INTERVAL_IN_MINUTES=1
```

```
# Relative path to system status info file.
# Example: logs/divaws.status
MONITORING_STATUS_FILE_RELATIVE_PATH=logs/divarchive_ws.status

# URL of monitoring web service.
# Example:
http://localhost:9763/services/Monitoring_1.0/MonitoringHttpSoap11En
dpoint
MONITORING_HTTP_ENDPOINT=http://localhost:9763/services/Monitoring_1
.0/MonitoringHttpSoap11Endpoint

#****************************************************************
# Logging
#****************************************************************

# Relative path to logging directory. Using by LoggerManager.
# Example: logs
LOGGING_DIRECTORY=logs

# Levels can be:  DEBUG, INFO, WARN, ERROR, FATAL
LOGGING_ROOT_LEVEL=INFO
LOGGING_TRACE_LEVEL=INFO
LOGGING_SERVICE_LEVEL=WARN

# File size should be specified using the convention:  #KB|MB
LOGGING_MAXFILESIZE=20MB

# All files (trace, service and .zip) older than this will be
removed
# hours
LOGGING_LIFETIME=50
```