opentext™

OpenText™ Output Transformation Server

Developer's Guide

This guide is intended for Output Transformation Server developers and provides information on developing with APIs and other more intermediate features.

VDTOTS240200-PGD-EN-1

OpenText™ Output Transformation Server
Developer's Guide
VDTOTS240200-PGD-EN-1
Rev.: 2024-Apr-16

This documentation has been created for OpenText™ Output Transformation Server CE 24.2.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

Introduction

This online help guide describes the OpenText Output Transformation Server engine, and is intended for Business Analysts and API developers.

1.1 Summary

The Output Transformation Server engine is the architecture and infrastructure for OpenText products. It allows the deployment of the high performance OpenText transformation products, events, and process flows to a JEE application server. Once deployed, the products can be run from several places: the events themselves; a session bean or SOAP/Web service; or interactively from OpenText Output Transformation Designer. The most basic example of this is to run a project when its input is copied to a specified directory.

The engine is responsible for running components that are either part of the core engine, specific to a OpenText product, or written by the OpenText Professional Services Group for specific client needs. Clients proficient in the Java programming language can also develop their own components. A common API enables the creation of both events and processes to ingest data and process data in the Output Transformation Server engine.

Components are broken down into two categories:

- **Service components**. Once started, a service component will run until some external force stops it. They typically listen for events to occur and perform some task. An Event is an example of a service component.
- **Runnable components**. Within a process flow, a runnable component is an element that performs some type of action. A runnable component can be classified as one of three general types:
  - **Router**. A component that routes data to different actions based on certain user-defined criteria. Examples include the TimeRouter or the XDocSizeRouter.
  - **Processor**. A component which handles the movement of information to return an output, such as the **Output Transformation Project** or **Data Transformation Project** components.
  - **Sender**. A component that transmits data to an outside source, like the file system or an FTP server. There are a few different processes that can handle this type of external communications; the FTPProcess and MailProcess are some examples.

Chapter 2

Developing in Output Transformation Server

Development of custom components in Output Transformation Server can help you adapt projects for your specific needs. For more information, consult the following topics to get started with developing in Output Transformation Server.

2.1 Javadocs

The *Javadoc* document provides a list of all Java packages and their associated class/interface descriptions. This document is an invaluable resource when developing your own components and is available from the **Help** menu in Output Transformation Designer.

2.2 Output Transformation Server Classpath

The Output Transformation Server classpath is where OpenText Output Transformation Designer loads the JAR files it requires. The following directories and their contents are added to the classpath upon starting Output Transformation Designer:

- `<install_home>\install\<version>\lib`
- `<install_home>\install\<version>\lib\clientOnly`
- `<install_home>\install\<version>\lib\common`
- `<install_home>\install\<version>\lib\endorsed`
- `<install_home>\install\<version>\dev-studio\plugins`
- `<install_home>\install\<version>\dev-studio\lib`
- `<install_home>\install\<version>\dev-studio\parserDB`

For example, Data Transformation Engine JAR files, such as `Xenos-t1transport.jar`, are loaded from the `<install_home>\install\<version>\lib\common` directory.

You can set the directories to the libraries with these JAR files by setting them with a classpath argument using a command prompt.

For example, if you wanted to compile source code for a Data Transformation Engine custom function outside of Output Transformation Designer, you might enter the following at the command prompt:

```
%JAVA_HOME%\javac -classpath ..\lib\Xenos-t1transform.jar;Xenos-framework-util.jar; Xenos-framework-engine.jar FUNCTION_NAME.java
```

where:

- `%JAVA_HOME%` is the directory in which your `javac.exe` is located.
- `FUNCTION_NAME` is the name of your custom function.

In addition to the above, any compiled classes can be placed under the `<install_home>\install\<version>\initialFiles\common\_classes` folder.

Note: If the Java classes are compiled within OpenText Output Transformation Designer, they will be picked up automatically at the above location. If they are compiled outside of Output Transformation Designer, a restart will be required for them to be recognized.

2.3 Sample Code

Various code samples of process flows and components are included with the application. Sample code can be found in the following location:

- `<install_home>\install\<version>\initialFiles\common\_sample\OutputTransformationServer`

The parameter sample codes are then separated into the folders for their respective parameter categories. For example, the sample code for a router is located in the `<install_home>\install\<version>\initialFiles\common\com\xenos\framework\bpengine\router` folder.

2.3.1 Events

Events are service components that wait for a particular user-specified incident to happen on the system. They can observe a watched directory or FTP site for files to be written, listen for incoming email or JMS messages, or await data to be delivered through a network socket.

For more information on events, see *OpenText Output Transformation Server - User Guide (VDTOTS-H-UGD)*. You can also find several samples of events located in the `<install_home>\install\<version>\initialFiles\common\_sample\OutputTransformationServer\event` directory.

2.3.1.1 HttpEvent

The HttpEvent component is designed to allow processes to be executed using web services. The event opens a port and waits for firewall-friendly HTTP requests.

The advantage to using the HttpEvent is that it is firewall friendly. The one drawback is that the input data is limited by your memory capacity, so it should be noted that this method works best with smaller files that can be held completely in memory.

Note: Client servlet(s) have to be created and deployed to handle HTTP requests.

Below, find the servlet code for the HttpEvent process that can be used as a basis for writing your own servlets:

```
package com.xenos.framework.event.http.servlet;

import java.io.*;
import java.util.*;
import javax.servlet.http.*;
import javax.servlet.*;

import org.apache.commons.fileupload.DiskFileUpload;
import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileUploadException;

import com.xenos.framework.event.http.*;
import com.xenos.framework.event.http.parm.*;
import com.xenos.framework.component.XDoc;
import com.xenos.framework.util.message.IXenosLogger;
import com.xenos.framework.GenericMessages;

public abstract class AbstractHttpServlet extends HttpServlet {

     protected static final String REQUEST_SOURCE_PARAM = "requestSource";
     protected static final String LOGGER = "framework.logger";
     protected static final String EVENT = "framework.httpEvent";
     protected static final String REQUEST_PARAMS = "framework.requestParams";
     protected DiskFileUpload m_file;
     protected String m_inputResource;
     public AbstractHttpServlet() {
     }

     /**
       * @see javax.servlet.GenericServlet#init(ServletConfig)
       */
     public void init(ServletConfig config) throws ServletException {
          super.init(config);
          m_inputResource = config.getInitParameter(REQUEST_SOURCE_PARAM);
          if (m_inputResource == null) {
               m_inputResource = RequestSourceEnum.SOCKET.toString();
          }
          if (RequestSourceEnum.HTML.equals(m_inputResource)) {
               m_file = new DiskFileUpload();
               m_file.setSizeMax(-1);
               m_file.setSizeThreshold(10 * 1024);
               m_file.setRepositoryPath(System.getProperty("java.io.tmpdir"));
          }
     }

     /**
      * @see javax.servlet.http.HttpServlet#doGet(HttpServletRequest,
HttpServletResponse)
      */
     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throwsServletException, IOException {
          resp.getOutputStream().println(printHTMLHeader() + "<body> <h3>Only can get
input from html FORM post!</h3></body></html>");
     }

       /**
       * @see javax.servlet.http.HttpServlet#doPost(HttpServletRequest,
HttpServletResponse)
       *
       * process uploaded files
       */
     public void doPost (HttpServletRequest req, HttpServletResponse res)
throwsServletException, IOException {
          IXenosLogger logger = (IXenosLogger)req.getSession().getAttribute(LOGGER);
          if (logger == null) {
               Object event = req.getSession().getAttribute(EVENT);
          }
          else if (event instanceof HttpEvent) {
```

```
               logger  = ((HttpEvent)event).getLogger();
          }
     }

     String[] params = (String[])req.getSession().getAttribute(REQUEST_PARAMS);

     if (logger == null) {
          res.getOutputStream().println(printHTMLHeader());
          res.getOutputStream().println("<body><h3>Can not get logger from session!</
h3></body></html>");
          return;
     }

     try {
          XDoc xdoc= new XDoc();
          OutputStream os = xdoc.obtainOutputStream();
          InputStream is = getInput(req, logger);
          long totalBytesRead = 0;
          byte[] rd = new byte[4 * 1024];
          for (int b = is.read(rd); b > -1; b = is.read(rd)) {
               os.write(rd, 0, b);
               totalBytesRead = totalBytesRead + b;
          }
          os.close();
          GenericMessages.issueDebug(logger, String.valueOf(xdoc.getSizeInBytes()) + "
bytes received OK!");

          Map reqParams = null;
          if (params != null) {
               reqParams = new HashMap();
               for (int i = 0; i < params.length; i ++) {
                    String param = req.getParameter(params[i]);
                    if (param != null) {
                         reqParams.put(params[i], param);
                    }
               }
          }

          XDoc response = process(req, res, xdoc, reqParams);

          if (response != null) {
               GenericMessages.issueDebug(logger, "Sending response...");
               InputStream resultStream = response.obtainInputStream();
               ServletOutputStream outStream = res.getOutputStream();
               long totalBytesWritten = pipeResponse(resultStream, outStream);
               GenericMessages.issueDebug(logger, String.valueOf(totalBytesWritten) + "
bytes response sent OK!");
          }

               GenericMessages.issueDebug(logger, getServletName() + " Servlet
 ...Processing request end");
     }
     catch (Throwable e) {
          GenericMessages.issueError(logger, e.getMessage());
          res.getOutputStream().println(printHTMLHeader());
          res.getOutputStream().println("<body><h3>" + e.toString() + "</h3></body></
html>");
     }
   }

     protected abstract XDoc process(HttpServletRequest req, HttpServletResponse res,
XDoc xdoc, Map params)
          throws Exception;

     protected InputStream getInput(HttpServletRequest req, IXenosLogger logger) throws
IOException, FileUploadException {
          InputStream is = null;
          if (RequestSourceEnum.SOCKET.equals(m_inputResource)) {
               Enumeration e = req.getHeaderNames();
               if (e != null) {
                    GenericMessages.issueDebug(logger, "========= HTTP request headers");
```

```
                    for (; e.hasMoreElements(); ) {
                         String name = (String)e.nextElement();
                         Enumeration en = req.getHeaders(name);
                         if (en != null) {
                              String value = "";
                              for (; en.hasMoreElements(); ) {
                                   value = value + (String)en.nextElement();
                              }
                              GenericMessages.issueDebug(logger, name + ": " + value);
                         }
                    }
                    GenericMessages.issueDebug(logger, "========= End of HTTP request
headers");
                }
          }
          is  = req.getInputStream();
     }
     else if (RequestSourceEnum.HTML.equals(m_inputResource)) {
          List fileItems = m_file.parseRequest(req);
          is = ((FileItem)fileItems.get(0)).getInputStream();
     }
     return is;
   }
   private long pipeResponse(InputStream is, ServletOutputStream os) throws IOException {
     byte[] buf = new byte[4 * 1024];
     int totalBytes = 0;
     try {
          if (is != null) {
               if (os == null) {
                    // just skip the bytes
                    is.skip(is.available());
               }
               else {
                    while(is.available() > 0) {
                         int bytesRead = is.read(buf);
                         os.write(buf, 0, bytesRead);
                         totalBytes = totalBytes + bytesRead;
                    }
               }
          }
     }
     catch (IOException ioe) {
          throw new IOException("Exception occurred getting HTTP response: " +
ioe.toString());
     }
     os.flush();
     return totalBytes;
   }

   /**
   * Generate default HTML header.
   */
   protected String printHTMLHeader () {
          return "<html><head><title>" + getServletName() + "</title></head>";
   }
}
```

2.3.1.2 Servlets

The following code is an example of the **EchoServlet** script:

```
package com.xenos.framework.event.http.servlet;
 import java.util.*;
 import javax.servlet.http.*;
 import com.xenos.framework.component.*;
 public class EchoServlet extends AbstractHttpServlet {

   public EchoServlet() {
        super();
    }

    protected XDoc process(HttpServletRequest req, HttpServletResponse
res, XDoc xdoc, Map params)
        throws Exception
    {
       return xdoc;
    }

 }
```

The following code is an example of the **HelloWorldServlet** script:

```
package com.xenos.framework.event.http.servlet;
 import java.util.Map;
 import javax.servlet.http.*;
 import com.xenos.framework.component.*;
 public class HelloWorldServlet extends AbstractHttpServlet {
    /**
     * Constructor for HelloWorldServlet.
     */
    public HelloWorldServlet() {
        super();
    }

```

```
    protected XDoc process(HttpServletRequest req, HttpServletResponse
res, XDoc xdoc, Map params) throws Exception {
        res.getOutputStream().println("<html><head><title>Demo</title>" +
                                      "</head><body><h1>Hello World</h1>");
        return null;
    }
}
```

2.3.2 Custom Components

The topics in this section discuss working with custom components:

2.3.2.1 Custom Component Wizard

If you require additional or customized functionality that is not provided by Output Transformation Server standard components, such as custom routing or integration with a back office system, you can create custom components. The **Custom Component Wizard** guides you through the steps for creating a skeleton Java class that you can fill in with your own custom business logic.

1. Create a **Custom Component** type new component. (For details about how to create a new component, see *OpenText Output Transformation Server - User Guide (VDTOTS-H-UGD)*.)

   The **Select Component Type** dialog opens.

2. Complete the following details about the component being created:

   - **Component Type**. Indicates the type of component being created. You can choose from either Runnable or Service; a **Runnable** component can be called from a process flow, while a **Service** component runs in the background to provide additional functionality to events and processes.
   - **File System**. Specifies the directory where the files will be created. Choose a mounted file systems from the drop-down list. If the directory you want to use is not listed, it is not mounted. For more information on mounting the file system, see *OpenText Output Transformation Server - User Guide (VDTOTS-H-UGD)*
   - **Package**. Specifies the Java package in which your Java file will reside. Choose a Java package from the drop-down list. Only the packages which currently reside in your mounted file system will be displayed in the drop-down list.
   - **Base Name**. Uniquely names the custom component. This value is required and must be entered without an extension in order to proceed with the wizard.

3. Click **Next**:

- If you selected **Runnable**, continue with "Configuring a Runnable Component" on page 16.
- If you selected **Service**, continue with see "Configuring a Service Component" on page 18.

2.3.2.1.1 Configuring a Runnable Component

The **Configure Runnable** dialog allows you to define the inputs, outputs and job variables required by this component.

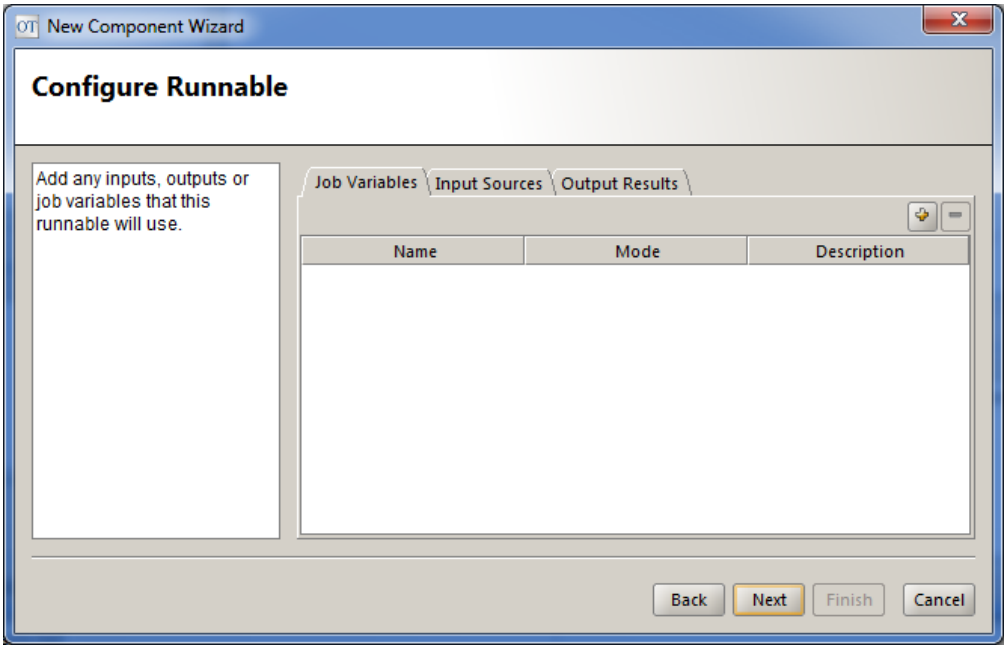

Figure 2-1: New Component Wizard's Configure Runnable screen

To add an item on this screen, click the **Add** button,  .

To remove an item, highlight the item and click the **Remove** button,  .

1. In the **Input Sources** section, enter a name and a description for each input source you wish to use for this component. You can create more than one input source, but each source must have a unique name. These values are optional.
2. In the **Output Results** section, enter a name and description for each output result you with to create from this component. You can create more than one output result, but each result must have a unique name. These values are optional.
3. In the **Job Variables** section, enter a name, mode and description for each variable required by this component. The possible values for **Mode** are:
   - **READ.** Indicates that the variable is required by this component.

- **WRITE**. Indicates that the variable is created by this component.
- **MODIFY**. Indicates that the variable can exist for this component, but can also be modified by this component.

You can create more than one variable, but each variable must have a unique name. These values are optional.

4. Click **Next** to open the **Configure Routing** dialog, which allows you to configure the routing for this component:

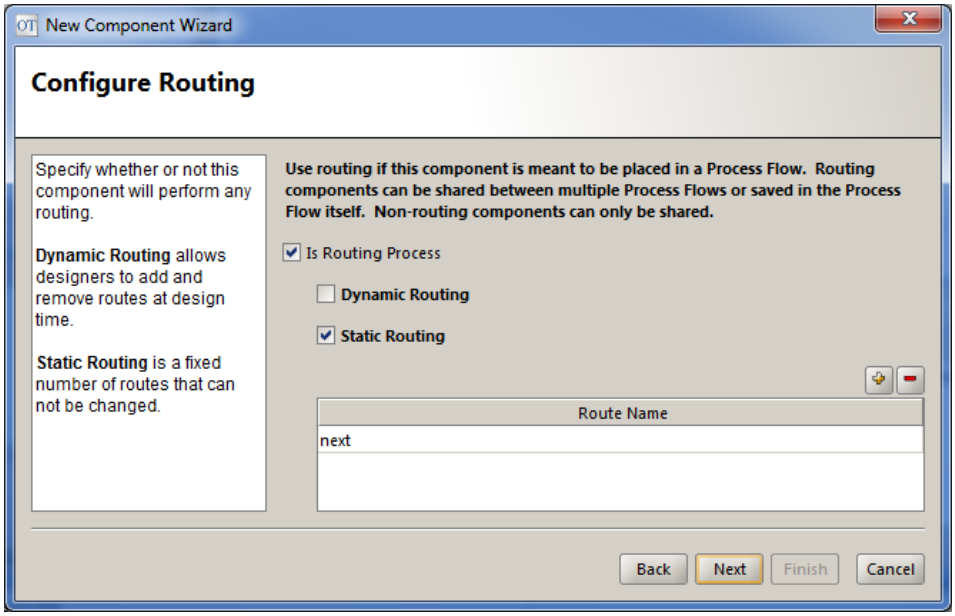

Figure 2-2: New Component Wizard's Configure Routing screen

5. Select the **Is Routing Process** option if you want this component to perform routing. Leave this option unselected if you want the component to have only one route.

   Selecting **Is Routing Process** enables the following additional routing options:

   - **Dynamic Routing.** Indicates that new routes can be created or removed from within the process flow.
   - **Static Routing**. Indicates that there is a predefined number of routes that cannot be modified. If selected, you can create these routes by clicking the **Add** button, , and entering a unique identifier under **Route Name**. You can remove an existing route by highlighting it and clicking the **Remove** button, .

   Dynamic and static routing are optional preferences, and are not mutually exclusive (i.e. you can select one, both, or neither of the options).

6. Select your routing preferences and click **Next** to "Configuring Parameter Objects" on page 19.

2.3.2.1.2 Configuring a Service Component

The **Configure Service** screen allows you to define the service type for this component.

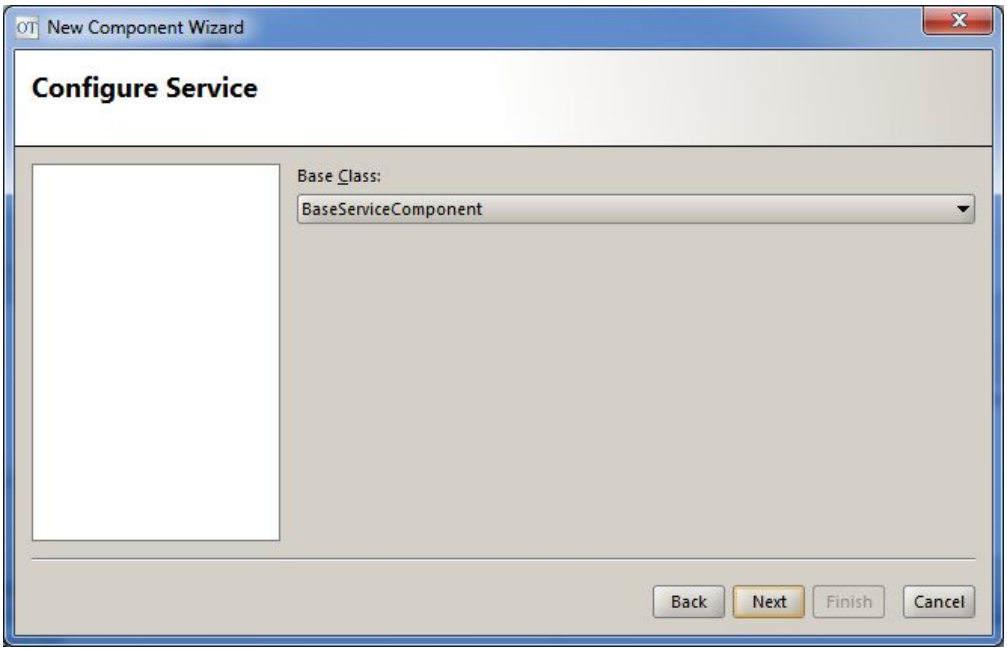

Figure 2-3: New Component Wizard's Configure Service screen

1. Use the drop-down list to select your **Base Class**. The available options are:

   - **BaseServiceComponent.** Indicates a service which runs in the background waiting for something to happen or communicate with it. This service does not submit any jobs to the Output Transformation Server engine.
   - **AbstractEvent.** Indicates a service which listens for something to happen and then submits a job to the Output Transformation Server engine.
   - **AbstractEventScanner.** Indicates a service which checks for something at predetermined times and then submits a job to the Output Transformation Server engine.

2. Click **Next** to continue with "Configuring Parameter Objects" on page 19.

2.3.2.1.3 Configuring Parameter Objects

The **Configure Parm Object** screen enables you to create parameter (parm) objects, which store configuration details for this component and which can be modified later in the generated .parmdef file. This means that you can create generic components whose configurable code allows for reuse.

To add a parm object on this screen, click the **Add** button  .

To remove a parm object, highlight the object and click the **Remove** button  .

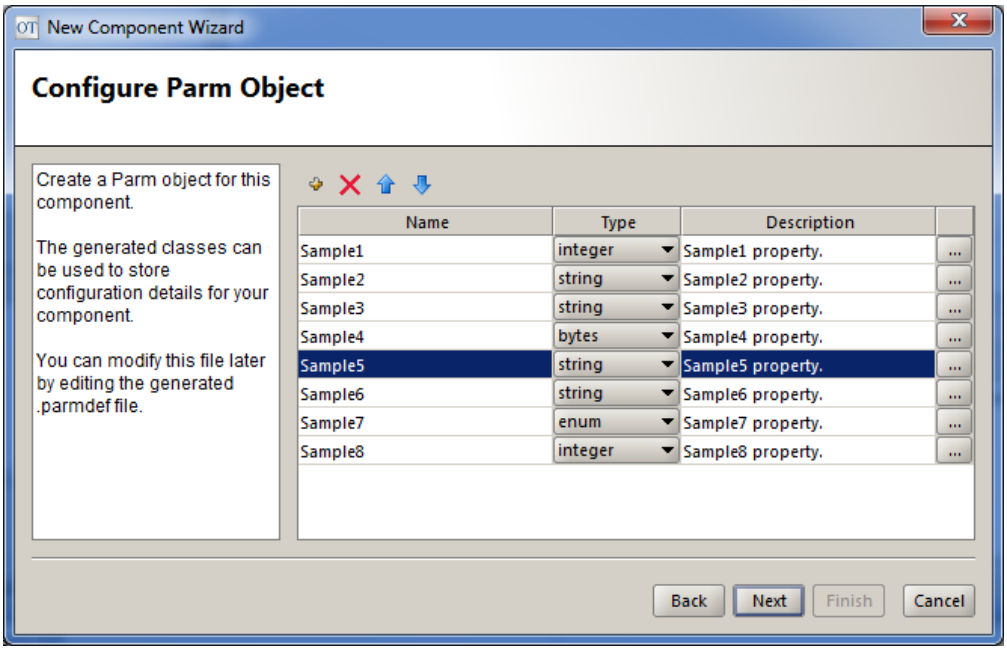

Figure 2-4: New Component Wizard's Configure Parm Object screen

Once you have added a parm object:

1. Enter a name for the parm object in the **Name** field. This value is required and must be unique.
2. Use the **Type** dropdown list to select the data type for the parm object. The available data types are:
   - **integer.** Specifies a whole number, such as 1, 2, or -5.
   - **float**. Indicates a number containing a decimal, such as 123.44.
   - **string**. Specifies a string of characters.
   - **boolean**. Indicates that the values will be True or False.
   - **enum**. Specifies a list of options, such as True, False, or fileNotFound.

- **object**. Specifies a parm object.

3. In the **Description** field, provide a description for the parm object. This value is optional.
4. Click the ... button to access advanced configuration settings which further define the parameters for the parm object value.
5. Click **Next** to open the **Configure Component Definition** dialog.

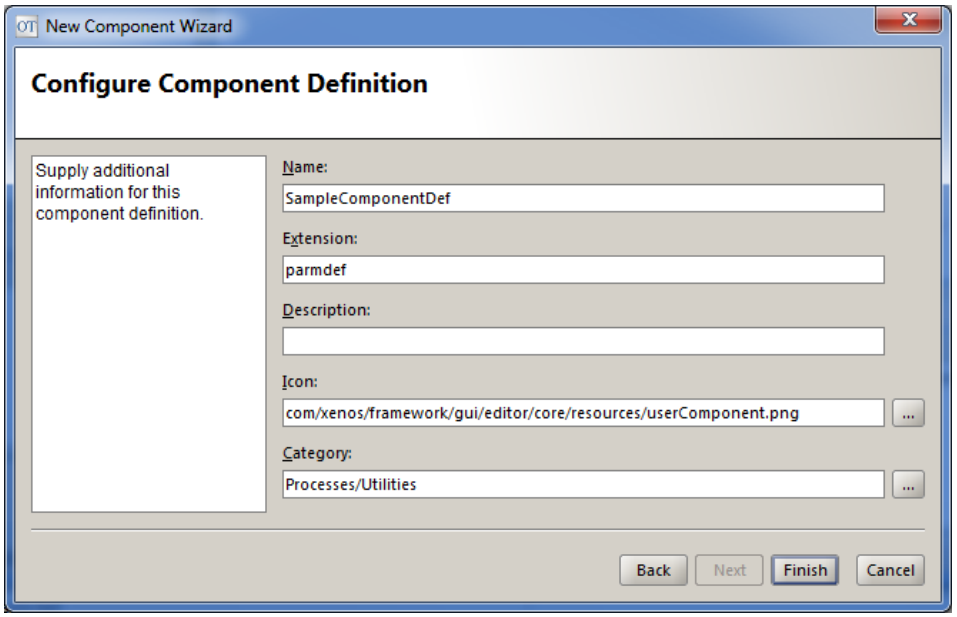

Figure 2-5: New Component Wizard's Configure Component Definition screen

6. Provide the following details about the component:
   - **Name.** Specifies the name associated with the component. This value is required.
   - **Extension.** Specifies the extension for the component name. This value is required.
   - **Description**. Lists a brief description of what the component does. This value is optional.
   - **Icon.** Specifies the classpath location of the component's icon for when it is viewed within a process flow. This value is optional.
   - **Category.** Specifies the category under which to place the component, entered in the `root/subCat/subCat...` format. This value is optional.
7. Click **Finish**.

The Java class is created in the file system at the location you specified. For more information about usage and method names, see "Output Transformation Server Classpath" on page 9.

Tip: Another way to configure a parmdef for a parm object is to run the Parmdef Editor, then define the parameters for a parm object in the **Properties** window. For more information, see "ParmDef Editor" on page 21.

2.3.2.2 ParmDef Editor

The ParmDef Editor (Parameter Definition Editor) is a Output Transformation Server component used to build or modify parm definitions for parm objects to add user properties to custom components. The **ParmDef Editor** appears under **Custom Components**.

In order to build parm objects, you first need to create the parm definition. Once you give a name and location to the parmdef, a `.parmdef` file is created. Use this file to generate the Java code for the parm object. Once this code is compiled, use it in the custom component it was created for. It can also be reused in other custom functions as required to minimize rework.

2.3.2.2.1 Creating a Parameter Definition for a Parameter Object

To create a parameter definition (parmdef) for a parameter (parm) object:

1. Run the ParmDef Editor , which is located under **Custom Components**, to open the **New Component Wizard** for the ParmDef Editor component.
2. Provide a **name** and **directory location** for the parmdef object and click **Finish to** create a new parmdef in the in the defined directory in the File System window. The name and directory will appear as parmdef properties in the **Properties window**.
3. Define or edit the parmdef class in the **Properties window**:

Class	Specifies each class's unique properties.
Style	Determines the type of parmdef to use: **standard** or **special**.
IsLicense	**True**, if this parm object has a license.

Note: You will find descriptions in the **Message** window for every parameter and their variables within the ParmDef Java class that you select.

4. Click the **Java** icon,  , to generate the parmdef source Java code and compile it for the parm object.

The compiled code can now be used in the custom component it was created for.

Tip: Another way to create a parmdef for a parm object is to configure the parm object using the **New Component Wizard**.

Related Links

- "ParmDef Editor" on page 21
- "Custom Component Wizard" on page 15
- *OpenText Output Transformation Server - User Guide (VDTOTS-H-UGD)*
- "Configuring Parameter Objects" on page 19

2.3.2.2.2 ParmDef Editor Toolbar

Icon	Menu Command	Description
	Add variable	Adds a variable to a ParmDef class array.
	Remove variable	Removes a variable from a ParmDef class array.
	Compile	Compiles the ParmDef Java code for variables to a ParmDef class array.

2.3.2.3 Runnable Definitions

You can customize the following code samples of runnable methods to get you started on creating your own runnable components.

2.3.2.3.1 Interface Definition (Runnable)

The following code is an example of the IRunableComponent script with supplementary information about certain passages of code embedded within:

```
package com.xenos.framework.component;

import java.util.*;

/**
 * Runnables are components that can be run via a JobTicket, and
 * have a finite task to do (They will end/stop on their own).
 */
public interface IRunableComponent extends IComponent {

   public void runComponent();

   /**
```

```
     * @return is an array describing any job variables used/created by this
component.
     */
    public JobVarDef[] getRequiredJobVars();

    /**
     * @return is an array describing any source streams used by this
component.
     */
    public SourceResultDef[] getRequiredSources();

    /**
     * @return is an array describing any result streams used/created by this
component.
     */
    public SourceResultDef[] getCreatedResults();
}
```

2.3.2.3.2 BaseComponent

The following code is an example of the BaseComponent script with supplementary information about certain passages of code embedded within. Users who wish to write their own runnables can extend BaseComponent.

```
package com.xenos.framework.component;

import java.util.*;
import com.xenos.d2e.parm.*;
import com.xenos.framework.util.message.*;
import com.xenos.framework.system.*;
import com.xenos.framework.environment.*;
import com.xenos.framework.logging.*;
import com.xenos.framework.exceptions.*;

/**
 * Abstract component.  Most components will extend this to get some basic
 * methods implemented automatically.
```

```
 */
public abstract class BaseComponent implements IComponent {

    public static final EventCode JOB_TICKET_ASSIGNED_EVENT = new
EventCode(0,"JobTicket has been assigned to the component.");
    public static final EventCode COMPONENT_INIT_EVENT = new EventCode(1,
"The component has been initialized.");
    public static final EventCode COMPONENT_START_EVENT = new EventCode(2,
"The component has started.");
    public static final EventCode COMPONENT_END_EVENT = new EventCode(3,
"The component has ended.");
    public static final EventCode PROCESSFLOW_PROCESS_STARTING_EVENT = new
EventCode(4,"The process in the process flow is starting.");
    public static final EventCode PROCESSFLOW_START_EVENT = new
EventCode(5, "The process flow has started.");
    public static final EventCode PROCESSFLOW_END_EVENT = new EventCode(6,
"The process flow is done.");
    public static final EventCode SUBPROCESSFLOW_START_EVENT = new
EventCode(7, "The sub process flow has started.");
    public static final EventCode SUBPROCESSFLOW_END_EVENT = new
EventCode(8, "The sub process flow is done.");

    protected ParmHandler m_componentParameters = null;
    protected String m_alternateParameters = null;
    protected IXenosLogger m_logger = null; // new NullLogger().
    protected ISystem m_system = null;
    protected IEnvironment m_environment = null;
    protected JobTicket m_jobTicket = null;
    protected int m_version_Id = 0;
    protected String m_componentName = "";
    protected String m_aliasName = "";
    private ArrayList m_allObservers = null; // Don't create until we know we
have Observers.

    /**
     * Called to pass in the component's parameters.  This is called
     * once before initialize() is called.
```

```
     *@param to the parm handler for this component.
     */
    public void configureParameters(ParmHandler po) {
       m_componentParameters = po;
    }

    /**
     * Retrieved the component's parms.
     *@return the parm handler for this component.
     */
    public ParmHandler getComponentParameters() {
       return m_componentParameters;
    }

    /**
     * Called once to set up the logger that this component will use
     * to issue messages/warning/errors.
     *@param logger is the logger to be used for this component when issuing
messages.
     */
    public void configureLogger(IXenosLogger logger) {
       m_logger = logger;
    }

    /**
     * Retrieves the active logger for this component.
     * @return the active logger for this component.
     */
    public IXenosLogger getLogger() {
       return m_logger;
    }

    /**
```

```
     * Called once to set up the system and from there the environment.
     *@param system is the system implementation being used.
     */
    public void configureSystem(ISystem system, IEnvironment environment)
{
        m_system = system;
        m_environment = environment;
    }

    /**
     * Retrieves the system from which this component was loaded.
     * @return the system which loaded this component.
     */
    public ISystem getSystem() {
        return m_system;
    }

    /**
     * Retrieves the environment.
     * @return the environment.
     */
    public IEnvironment getEnvironment() {
        return m_environment;
    }

    /**
     * Retrieves the JobTicket, which may be null.
     * @return the job ticket or null.
     */
    public JobTicket getJobTicket() {
        return m_jobTicket;
    }
```

```
    /**
     * Retrieves the environment.
     * @return the environment.
     */
    public void setJobTicket(JobTicket jt) {
       m_jobTicket = jt;
       if (getComponentParameters() != null) {
          getComponentParameters().setPoJobVarResolver(jt.
getJobVarResolver());
       }

       List observers = jt.getAllComponentObservers();
       for(int i=0;i<observers.size();i++) {
          addObserver((IComponentObserver) observers.get(i));
       }
       IXenosLogListener logListener = jt.getJobTicketLogListener();
       IXenosLogger compLogger = getLogger();
       if (compLogger instanceof XenosMasterLogger) {
          synchronized (compLogger) {
             if (!
((XenosMasterLogger)compLogger).getListeners().contains(logListener)) {
                ((XenosMasterLogger)compLogger).addLogListener(logListen
er);
             }
          }
       }
       else {
          throw new GenericRuntimeException("Falied to add jobTicket log
listener  to component "+getComponentName());
       }

       issueEvent(JOB_TICKET_ASSIGNED_EVENT, this);

    }
```

```
    /**
     * Called once before this component is used (ran, started or otherwise
     * used by other components).
     */
    public abstract void initialize();

    /**
     * Adds a class that wants to observe any status changes to this
component.
     * @param Observer is the class wanting to keep an eye on any status
updates.
     */
    public void addObserver(IComponentObserver observer) {
       lazyCreateObserverVector();
       if (observer != null) {
          synchronized(m_allObservers) {
             // TS This is probably overkill
             if (! m_allObservers.contains(observer)) {
                m_allObservers.add(observer);
             }
          }
       }
    }

    /**
     * Removes a class previously added with addObserver().
     * @param Observer is the class that no longer wants status updates.
     */
    public void removeObserver(IComponentObserver observer) {
       if (observer != null && m_allObservers != null) {
          synchronized(m_allObservers) {
             for (int i=0; i< m_allObservers.size(); i++) {
                if (observer == m_allObservers.get(i)) {
```

```
                    m_allObservers.remove(i);
                    break;
                }
            }
        }
    }

    /**
     * Clears all Observers of this component.
     */
    public void clearObservers() {
        if (m_allObservers != null) {
            synchronized(m_allObservers) {
                m_allObservers.clear();
            }
        }
    }

    /**
     * Notifies all the observers that the status has changed.  Any time the component
     * updates its status, this method should be called.
     *@param status is the status object for this component.
     */
    public void notifyObservers(ComponentStatus status) {
        if (m_allObservers != null) {
            synchronized(m_allObservers) {
                // ??? Should I get an array so I don't have to keep a lock on the array during the loop
                // ??? Should I get an array so I don't have to keep a lock on the array during the loop
                // ??? Should I get an array so I dont have to keep a lock on the array during the loop
```

```
            for (int i=0; i< m_allObservers.size(); i++) {
                IComponentObserver observer = (IComponentObserver)m_
allObservers.get(i);
                observer.statusHasChanged(status);
            }
         }
      }

      /**
       * Issue an event to all observers.
       * @param eventCode is one of the component's static EventCodes
       * @param obj is null or an Object that supports the event.
       */
      public void issueEvent(EventCode eventCode, Object obj) {
         if (m_allObservers != null) {
            synchronized(m_allObservers) {
               // ??? Should I get an array so I don't have to keep a lock on
the array during the loop
               // ??? Should I get an array so I don't have to keep a lock on
the array during the loop
               // ??? Should I get an array so I don't have to keep a lock on
the array during the loop
               for (int i=0; i< m_allObservers.size(); i++) {
                  IComponentObserver observer = (IComponentObserver)m_
allObservers.get(i);
                  observer.issueEvent(eventCode, obj);
               }
            }
         }
      }

      /**
       * Helper method to synchronize the creation of the list of Observers,
setting
```

```
     * m_allObservers.
     */
    private synchronized void lazyCreateObserverVector() {
        if (m_allObservers == null) {
            m_allObservers = new ArrayList();
        }
    }
    public void setVersionId(int id) {
        m_version_Id = id;
    }
    public int getVersionId() {
        return m_version_Id;
    }

    /**
     * Called to pass in the components parameters.  This, or
     * configureParameters, is called
     * once before initialize() is called.
     * @param configFile is the absolute file of the config parameters when
the
     * parameters are not a parm object.
     */
    public void configureAlternateParameters(String configFile) {
        m_alternateParameters = configFile;
    }

    /**
     * Returns a componentName of a created instance.
     */
    public String getComponentName() {
        return m_componentName;
    }

```

```
    /**
     * Sets a componentName of a created instance.
     */
    public void setComponentName(String name) {
       m_componentName= name;
    }

    /**
     * Returns an aliasName of a created instance.
     */
    public String getAliasName() {
       return m_aliasName;
    }

    /**
     * Sets an aliasName of a created instance.
     */
    public void setAliasName(String name) {
       m_aliasName= name;
    }

    /**
     * Cleans all resources assigned to the component.
     * Resets jobTicket to null, and removes all observers.
     */
    public void jobHasEnded(){
       clearObservers();
       if (m_jobTicket!=null) {
          IXenosLogListener logListener = m_jobTicket.
getJobTicketLogListener();
          IXenosLogger compLogger = getLogger();
          if (compLogger instanceof XenosMasterLogger) {
             synchronized (compLogger) {
```

```
                if
 (((XenosMasterLogger)compLogger).getListeners().contains(logListener))
{

                   ((XenosMasterLogger)compLogger).removeLogListener(lo
gListener);

                }

             }

             m_jobTicket=null;

          } else {

             m_jobTicket=null;

             throw new GenericRuntimeException("Falied to remove
jobTicket log listener from component "+getComponentName());

          }

       }

    }

 }
```

2.3.2.3.3 Custom Processes

In order to create custom processes, you need to extend the BaseComponent script, and additionally, implement the IProcess interface. The following code is a sample of IProcess with supplementary information about certain passages of code embedded within. For more information, see "BaseComponent" on page 23.

```
package com.xenos.framework.bpengine;

import java.util.*;
import com.xenos.d2e.parm.*;
import com.xenos.framework.util.message.*;
import com.xenos.framework.component.*;
import com.xenos.framework.bpengine.parm.*;

/**
 * Interface that all Processes must implement.
 */
public interface IProcess extends IRunableComponent {

```

```
    /**
     * Returns the number of routes this process has.  If this process
performs
     * dynamic routing then implementors must include all
     * {@link BPConditionParm}s.
     */
    public int getNumberOfOutput();

    /**
     * Gets the route name based on the specified index.  If this process
     * performs dynamic routing then implementors must search through their
     * lists of {@BPConditionParm}s as well.
     */
    public String getOutputName(int index);

    /**
     * Determines whether this process can have dynamic routes.  A dynamic
route
     * is a new route or output that is added at design time.  A route or
output
     * is stored in the process parm as either a {@link BPConditionParm} or a
     * subclass of BPConditionParm.
     */
    public boolean canCreateNewOutput();

    /**
     * Removes the route at the specified index.  Processes that perform
     * dynamic routing should remove the corresponding {@link
BPConditionParm}
     * from the process parm.
     */
    public boolean removeOutput(int index);

    /**
```

```
     * Determines whether or not the route or output at the specified index
     * can be removed.
     * @see removeOutput}
     */
    public boolean canRemoveOutput(int index);

    /**
     * Implementors should create an instance of their own condition object
and
     * return it.  The condition should also be added to the process parm
before
     * returning.
     */
    public BPConditionParm createNewOutput();

    /**
     * Returns the condition at the specified index.
     */
    public BPConditionParm getOutputCondition(int index);
    /**
     * Causes the work for this process to be done, the return
     * value is the output number (index) which the flow should
     * move to.
     */
    public int performProcess(/*some args may be needed*/);
 }
```

2.3.2.4 Services Definitions

You can customize the following code samples of service methods to get you started on creating your own service components.

2.3.2.4.1 Interface Definition (Services)

The following code is an example of the BaseServiceComponent script. If you are looking to write your own service component, you can extend the BaseServiceComponent to help meet your requirements.

```
package com.xenos.framework.component;

import com.xenos.framework.exceptions.GenericRuntimeException;
import com.xenos.d2e.exceptions.*;

/**
 * Abstract service component.
 * Most service components will extend this to get some basic
 * methods implemented automatically.
 */

public abstract class BaseServiceComponent extends BaseComponent implements
IServiceComponent {
    private Object m_lock= new Object();
    private boolean m_locked= false;
    protected boolean m_started= false;
    protected boolean m_restartable = true;
    protected boolean m_clusterSingleton = false;
    public BaseServiceComponent() {
    }

    public void initialize() {
    }
    public boolean isStarted() {
        return m_started;
    }
    public void setStatus(boolean status) {
        m_started=status;
    }
```

```
    public final void startComponent(){
        setStatus(true);
        m_jobTicket.setStatusJobIsNotDone();
        m_jobTicket.setStatus(JobTicket.RUNNING);
        m_locked= true;
        try {
            startService();
        }
        catch (Throwable throwable) {
            getLogger().log(throwable);
            setStatus(false);
            m_jobTicket.setStatus(JobTicket.ERRORED);
            m_jobTicket.setFailureMessage(ExceptionUtil.getFullString(throwable));
            m_jobTicket.setStatusJobIsDone();
            getSystem().shutDownService(getComponentName());
        }
        finally {
            synchronized (m_lock){
                m_locked= false;
                m_lock.notify();
            }
        }
    }

    public final void stopComponent(){
        if (m_started) {
            try {
                stopService();
                synchronized (m_lock){
                    while (m_locked){
                        try {
                            m_lock.wait();
                        } catch (InterruptedException e) {
                            m_locked= false;
                        }
```

```
                    }
                }
            }
            catch (Throwable throwable) {
                getLogger().log(throwable);
            }
            finally {
                m_jobTicket.setStatus(JobTicket.FINISHED);
                m_jobTicket.setStatusJobIsDone();
                setStatus(false);
            }
        } else {
            throw new GenericRuntimeException("Service "+m_jobTicket.getComponentName()
+" has not been started.");
        }
    }
    public void setRestartable(boolean restartable) {
        m_restartable= restartable;
    }
    public boolean isRestartable(){
        return m_restartable;
    }
    public boolean isRegistered(){
        return false;
    }
    public boolean isClusterSingleton() {
        return m_clusterSingleton;
    }
    public void setClusterSingleton(boolean status) {
        m_clusterSingleton = status;
    }
    public abstract void startService();
    public abstract void stopService();
    public abstract boolean needsThread();
}
```

2.3.2.4.2 Service Components

The following code is an example of the IServiceComponent script with supplementary information about certain passages of code embedded within.

```
package com.xenos.framework.component;

/**
 * Interface that all services running in the Xenos Framework must
implement.
 */

public interface IServiceComponent extends IComponent {
    public void startComponent();
    public void stopComponent();
    public boolean isStarted();
    public void setStatus(boolean status);
    public boolean needsThread();
    public boolean isRestartable();
    public boolean isRegistered();
    public void setRestartable(boolean restartable);
    public boolean isClusterSingleton();
    public void setClusterSingleton(boolean status);
}
```

Chapter 3

JobRunner Web Service

The **JobRunner** is a set of APIs used to run a component. There are several ways to have Output Transformation Server run a job:

- Session Bean
- Web Service
- Direct

This section focuses on using the Job Runner web service interface.

3.1 JobRunner Web Service Interface

Running web service jobs through the Job Runner requires the `RunWebServiceJob.bat` file. This file is installed as part of the initial installation process in `<install_home>\sampleapplication\bin\bootstrap`.

In addition, when adding a custom application that will interface with the Job Runner Web Service, you will require the following .jar files, which are included in the OpenText Output Transformation Server installation:

For compile:

- Xenos-framework-client.jar
- Xenos-framework-engine.jar

For runtime:

- Xenos-framework-util.jar
- Xenos-framework-j2eeBridge.jar (located in `<install_home>\dynamiclib`)
- Xenos-framework-j2ee.jar
- jaxrpc.jar
- commons-logging.jar
- commons-discovery-0.2.jar
- saaj-api.jar

Note: Unless otherwise noted, the specified files are located in the `<install_home>\install\<version>\lib\common` directory.

The Job Runner web service interface has only one method: the **runJob** method. This method requires 4 input parameters and returns a single result.

3.1.1 Input

Parameter (Type)	Description
In0 (String)	**Configuration File**. The relative path to the configuration file to be run. Any IRunable component can be executed in this fashion. A sample process flow could be executed using this value: `_sample/OutputTransformationServer/processFlow/Isv2File.xProcessFlow`
In1 (MapData)	**Variable Map**. A key value pair that contains variables to be used by IRunable components.
In2 (MapData)	**InputMap**. A key value pair that typically contains input data to be read by (but not limited to) IRunable components.
In3 (MapData)	**ResultMap**. A key value pair that typically contains result data that is written by (but not limited to) IRunable components.

3.1.2 Result

Parameter (Type)	Description
runJobReturn (JobTicketWrapper)	A complex data type that contains information about the job execution. See the **JobTicketWrapper** table below for detailed information.

3.1.3 JobTicketWrapper

Field	Description
<*Alias*>	A name associated with the configuration file path. This alias is configured through the component pool.
<*Component*>	The relative path to the configuration file.
<*JobID*>	A unique ID given to the job execution.
<*Status*>	Status of job execution. For example, FINISHED, ABORTED, ERRORED.
<*TotalMS*>	Total run time, in milliseconds, for the job execution. Total time = RunningMS + QueuedMS + SuspendMS.
<*QueuedMS*>	The amount of time this job was in the queue.

Field	Description
<SuspendMS>	The amount of time this job was in the SUSPENDED state. Jobs can be suspended if a component is an IPausable component.
<RunningMS>	The amount of time this job was running.
<InputSize>	The size of the input data in bytes.
<OutputSize>	The size of the result data in bytes.
<StartTime>	A timestamp of when the job was started.
<StopTime>	A timestamp of when the job was completed.
<InvocationMethod>	Typically the alias of the event that triggered the job execution.
<UserField>	Typically unused but can be populated by any custom user IRunable component.
<FailureMessage>	Typically the exception that caused job execution to fail.
<WarningNumber>	The number of warnings encountered during job execution.
<ErrorNumber>	The number of errors encountered during job execution.
<FatalErrorNumber>	The number of fatal errors encountered during job execution.
<InputMap>	The returned input map after job execution. This map may be modified by the IRunable component.
<ResultMap>	The returned result map after job execution. This map will usually contain a result (1 or more) produced by the IRunable component.

Note: For more information about the complex data types, see "JobRunner WSDL" on page 45.

For more information about the JobTicketWrapper, see "Sample JobRunner Code" on page 53.

3.2 Authentication

Basic authentication is used where authentication is handled in the HTTP layer. This assumes that the connection between the client and the server will already be trusted and secure.

The following is a sample of the HTTP header sent during a SOAP request:

```
 POST /ws/services/JobRunnerWebService HTTP/1.1
 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol
2.0.50727.1433)
 AUTHORIZATION: Basic Sm9iUnVubmVyOkpvYlJ1bm5lcg==
 Content-Type: text/xml; charset=utf-8
 SOAPAction: ""
 Host: localhost:8080
 Content-Length: 4469
 Expect: 100-continue
 Connection: Keep-Alive
```

Note: The default user:pass is JobRunner:JobRunner.

3.3 Returning Custom Job Variables

When creating Output Transformation Server projects, custom variables can be added to suit your requirements. These variables will be returned to the calling process, be it a WebService, or the OpenText JobRunner.

To write the variables to a file when running a process through the JobRunner, modify `RunJob.argfile`:

1. Open the `RunJob.argfile` located in:

   `<install_home>\sampleApplication\ProcessManger\bin`

2. Specify any job variables and where they will be written to as in the following example:

```
////////////////////////////////
//
// Specify any job variables:
//
// -jobvar=x=5 - The variable x will contain the value 5 when the project starts.
//
//
-jobvar=somevar=somevalue
// Specify where the job variables are written:
//
// -jobvaroutputfile=filename.ext - The job variables will be written to this file.
-jobvaroutputfile=jobVarOutput.txt
```

The results will be written to the file you specified.

3.4 JobRunner WSDL

The WSDL document for the Job Runner web service interface is located here:

`http://localhost:8080/ws/services/JobRunnerWebService?wsdl`

The contents of the WSDL document are as follows:

```
 <?xml version="1.0" encoding="utf-8"?>
 <wsdl:definitions xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:tns1="http://
serialization.j2ee.xenos.com" xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:impl="http://
jobrunner.ws.framework.xenos.com" xmlns:intf="http://jobrunner.ws.framework.xenos.com"
targetNamespace="http://jobrunner.ws.framework.xenos.com" xmlns:wsdl="http://
schemas.xmlsoap.org/wsdl/">
   <wsdl:types>
     <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://
jobrunner.ws.framework.xenos.com">
       <xsd:import namespace="http://serialization.j2ee.xenos.com" />
       <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
       <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
       <xsd:complexType name="ArrayOf_xsd_string">
         <xsd:complexContent mixed="false">
           <xsd:restriction base="soapenc:Array">
             <xsd:attribute wsdl:arrayType="xsd:string[]" ref="soapenc:arrayType" />
           </xsd:restriction>
         </xsd:complexContent>
       </xsd:complexType>
       <xsd:complexType name="ArrayOf_xsd_anyType">
         <xsd:complexContent mixed="false">
           <xsd:restriction base="soapenc:Array">
             <xsd:attribute wsdl:arrayType="xsd:anyType[]" ref="soapenc:arrayType" />
           </xsd:restriction>
         </xsd:complexContent>
       </xsd:complexType>
     </xsd:schema>
     <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://
serialization.j2ee.xenos.com">
       <xsd:import namespace="http://jobrunner.ws.framework.xenos.com" />
       <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
```

```
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
      <xsd:complexType name="MapData">
        <xsd:sequence>
          <xsd:element name="keys" nillable="true" type="intf:ArrayOf_xsd_string" />
          <xsd:element name="values" nillable="true" type="intf:ArrayOf_xsd_anyType" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="JobTicketWrapper">
        <xsd:sequence>
          <xsd:element name="alias" nillable="true" type="xsd:string" />
          <xsd:element name="component" nillable="true" type="xsd:string" />
          <xsd:element name="errorNumber" type="xsd:long" />
          <xsd:element name="failureMessage" nillable="true" type="xsd:string" />
          <xsd:element name="fatalErrorNumber" type="xsd:long" />
          <xsd:element name="inputMap" nillable="true" type="tns1:MapData" />
          <xsd:element name="inputSize" type="xsd:long" />
          <xsd:element name="invocationMethod" nillable="true" type="xsd:string" />
          <xsd:element name="jobId" nillable="true" type="xsd:string" />
          <xsd:element name="outputSize" type="xsd:long" />
          <xsd:element name="queuedMs" type="xsd:long" />
          <xsd:element name="resultMap" nillable="true" type="tns1:MapData" />
          <xsd:element name="runningMs" type="xsd:long" />
          <xsd:element name="startTime" nillable="true" type="xsd:dateTime" />
          <xsd:element name="status" nillable="true" type="xsd:string" />
          <xsd:element name="stopTime" nillable="true" type="xsd:dateTime" />
          <xsd:element name="suspendMs" type="xsd:long" />
          <xsd:element name="totalMs" type="xsd:long" />
          <xsd:element name="userField" nillable="true" type="xsd:string" />
          <xsd:element name="warningNumber" type="xsd:long" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="runJobRequest">
    <wsdl:part name="in0" type="xsd:string" />
```

```
    <wsdl:part name="in1" type="tns1:MapData" />
    <wsdl:part name="in2" type="tns1:MapData" />
    <wsdl:part name="in3" type="tns1:MapData" />
  </wsdl:message>
  <wsdl:message name="runJobResponse">
    <wsdl:part name="runJobReturn" type="tns1:JobTicketWrapper" />
  </wsdl:message>
  <wsdl:portType name="JobRunnerWebService">
    <wsdl:operation name="runJob" parameterOrder="in0 in1 in2 in3">
      <wsdl:input name="runJobRequest" message="intf:runJobRequest" />
      <wsdl:output name="runJobResponse" message="intf:runJobResponse" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="JobRunnerWebServiceSoapBinding" type="intf:JobRunnerWebService">
    <wsdlsoap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc" />
    <wsdl:operation name="runJob">
      <wsdlsoap:operation soapAction="" />
      <wsdl:input name="runJobRequest">
        <wsdlsoap:body use="encoded" namespace="http://
jobrunner.ws.framework.xenos.com" encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/" />
      </wsdl:input>
      <wsdl:output name="runJobResponse">
        <wsdlsoap:body use="encoded" namespace="http://
jobrunner.ws.framework.xenos.com" encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="JobRunnerWebServiceService">
    <wsdl:port name="JobRunnerWebService" binding="intf:JobRunnerWebServiceSoapBinding">
      <wsdlsoap:address location="http://localhost:8080/jobrunner/services/
JobRunnerWebService" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

3.5 JobRunner Sample Request

The following is a sample SOAP request generated while trying to execute a sample Data Transformation Engine project:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="http://jobrunner.ws.framework.xenos.com"
xmlns:types="http://jobrunner.ws.framework.xenos.com/encodedTypes"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<tns:runJob>
   <configFile xsi:type="xsd:string">
      _sample/DataTransformation/pdf/XML to PDF.mgp
   </configFile>
   <varMap href="#id1" />
   <inputMap href="#id2" />
   <resultMap href="#id3" />
</tns:runJob>
<q1:MapData id="id1" xsi:type="q1:MapData"
   xmlns:q1="http://serialization.j2ee.xenos.com">
   <keys href="#id4" />
   <values href="#id5" />
</q1:MapData>
<q2:MapData id="id2" xsi:type="q2:MapData"
      xmlns:q2="http://serialization.j2ee.xenos.com">
      <keys href="#id6" />
      <values href="#id7" />
</q2:MapData>
<q3:MapData id="id3" xsi:type="q3:MapData"
   xmlns:q3="http://serialization.j2ee.xenos.com">
   <keys xsi:nil="true" />
   <values xsi:nil="true" />
```

```
</q3:MapData>
<soapenc:Array id="id4" soapenc:arrayType="xsd:string[1]">
   <Item>ext</Item>
</soapenc:Array>
<soapenc:Array id="id5" soapenc:arrayType="xsd:anyType[1]">
   <Item xsi:type="xsd:string">pdf</Item>
</soapenc:Array>
<soapenc:Array id="id6" soapenc:arrayType="xsd:string[1]">
   <Item>DataToConvert</Item>
</soapenc:Array>
<soapenc:Array id="id7" soapenc:arrayType="xsd:anyType[1]">
   <Item xsi:type="xsd:base64Binary">
   ... Data Removed ...
   </Item>
</soapenc:Array>
</soap:Body>
</soap:Envelope>
```

3.6 JobRunner Sample Response

The following is a sample SOAP response returned after the sample Data Transformation Engine project was executed:

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<ns1:runJobResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:ns1="http://jobrunner.ws.framework.xenos.com">
   <runJobReturn href="#id0" />
</ns1:runJobResponse>
<multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
   xsi:type="ns2:JobTicketWrapper"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:ns2="http://serialization.j2ee.xenos.com">
```

```
    <alias xsi:type="soapenc:string"></alias>
    <component xsi:type="soapenc:string"> _sample/DataTransformation/pdf/XML to PDF.mgp
    </component>
    <errorNumber href="#id1" />
    <failureMessage xsi:type="soapenc:string"></failureMessage>
    <fatalErrorNumber href="#id2" />
    <inputMap href="#id3" />
    <inputSize href="#id4" />
    <invocationMethod xsi:type="soapenc:string"></invocationMethod>
    <jobId xsi:type="soapenc:string">20081111_1034_001</jobId>
    <outputSize href="#id5" />
    <queuedMs href="#id6" />
    <resultMap href="#id7" />
    <runningMs href="#id8" />
    <startTime xsi:type="xsd:dateTime">
      2008-11-11T15:34:31.352Z
    </startTime>
    <status xsi:type="soapenc:string">FINISHED</status>
    <stopTime xsi:type="xsd:dateTime">
      2008-11-11T15:34:33.684Z
    </stopTime>
    <suspendMs href="#id9" />
    <totalMs href="#id10" />
    <userField xsi:type="soapenc:string"></userField>
    <warningNumber href="#id11" />
 </multiRef>
 <multiRef id="id7" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="ns3:MapData"
    xmlns:ns3="http://serialization.j2ee.xenos.com"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    <keys soapenc:arrayType="xsd:string[1]"
       xsi:type="soapenc:Array">
       <keys xsi:type="xsd:string">Product_Report</keys>
    </keys>
```

```
    <values soapenc:arrayType="xsd:anyType[1]"
       xsi:type="soapenc:Array">
       <values xsi:type="soapenc:base64Binary">
         ... Data Removed ...
       </values>
    </values>
 </multiRef>
 <multiRef id="id3" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="ns4:MapData"
    xmlns:ns4="http://serialization.j2ee.xenos.com"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    <keys soapenc:arrayType="xsd:string[1]"
       xsi:type="soapenc:Array">
       <keys xsi:type="xsd:string">DataToConvert</keys>
    </keys>
    <values soapenc:arrayType="xsd:anyType[1]"
       xsi:type="soapenc:Array">
       <values xsi:type="soapenc:base64Binary">
          ... Data Removed ...
       </values>
    </values>
 </multiRef>
 <multiRef id="id2" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    0
 </multiRef>
 <multiRef id="id1" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    0
 </multiRef>
 <multiRef id="id10" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
```

```
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    2332
 </multiRef>
 <multiRef id="id9" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    0
 </multiRef>
 <multiRef id="id8" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    2332
 </multiRef>
 <multiRef id="id5" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    295562
 </multiRef>
 <multiRef id="id11" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    0
 </multiRef>
 <multiRef id="id4" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
    2199
 </multiRef>
 <multiRef id="id6" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
    xsi:type="xsd:long"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
```

```
    0
</multiRef>
</soapenv:Body>
</soapenv:Envelope>
```

3.7 Sample JobRunner Code

The Job Runner web interface was designed to be interoperable. The web interface can be invoked by any type of client.

The following is a sample of how a C# client can call the runJob method.

3.7.1 Scenario

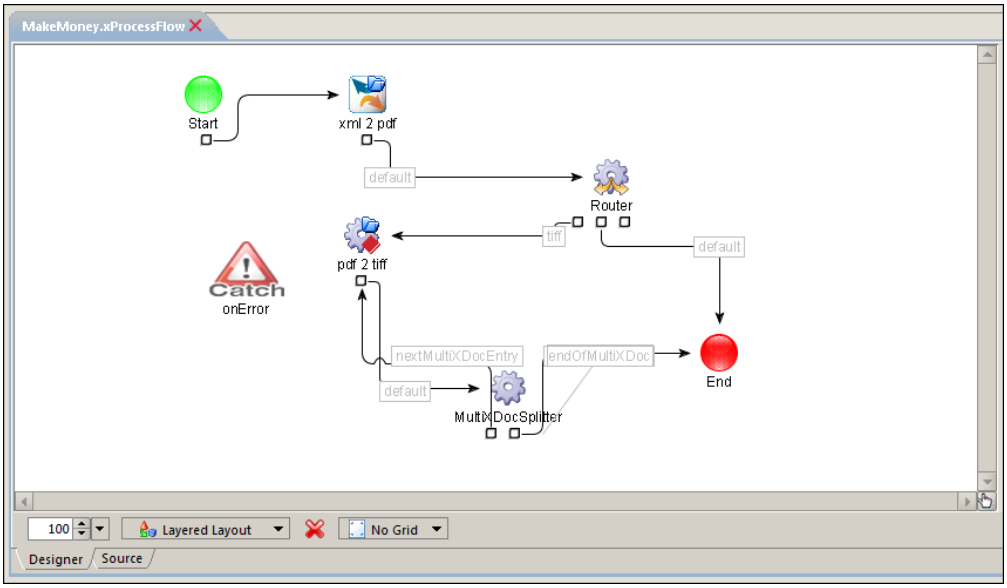

Figure 3-1: Sample process flow

A process flow is located on the server and is identified by the configuration file `XYZCorp/process/MakeMoney.xProcessFlow`. The `MakeMoney.xProcessFlow` is a process flow that will run an XML to PDF transform, and then do routing based on a job variable called `ext`. If the value of `ext` is `pdf`, then a flattened PDF form is returned. If the value of `ext` is `tiff`, then another PDF to TIFF transform will be used in order to return a TIFF image.

3.7.1.1 Code

The following is a sample client written using C#. This code sends an XML file to the server, along with a job variable which defines the format of the result.

```
// setup parameters for JobRunner web service
string configFile = "XYZCorp/process/MakeMoney.xProcessFlow";
MapData inputMap = new MapData();
MapData resultMap = new MapData();
MapData varMap = new MapData();

// read input data
byte[] inputDoc = File.ReadAllBytes(
    "C:/OpenText/OutputTransformationServer/sampleApplication/bin/Input.xml");
inputMap.keys = new string[] { "DataToConvert" };
```

```
inputMap.values = new Object[] { inputDoc };

// define extension variable as pdf return type
varMap.keys = new string[] { "ext" };
varMap.values = new Object[] { "pdf" };

// create the JobRunner web service client
JobRunnerWebServiceService service = new JobRunnerWebServiceService(
    "localhosts", "8080", "JobRunner", "JobRunner");

// run the job
JobTicketWrapper jt = service.runJob(configFile, varMap,
    inputMap, resultMap);

// print out some statistics to the console
Console.WriteLine("     Job Status: " + jt.status);
Console.WriteLine(" Total Run Time: " + jt.totalMs + " ms");
Console.WriteLine("Failure Message: " + jt.failureMessage);
Console.WriteLine("");

// write all result data to temp directory
MapData results = jt.resultMap;
string[] keys = results.keys;
object[] values = results.values;
for (int x = 0; x < keys.Length; x++) {
    string key = keys[x];
    Console.WriteLine("Found result: " + key);
    byte[] bytes = (byte[]) values[x];
    File.WriteAllBytes("c:/temp/" + key + "." +
        varMap.values[0], bytes);
}
Console.Write("Press any key to continue . . . ");
Console.ReadKey(true);
```

Chapter 4

Using Swagger UI and Output Transformation Server REST API Resources

Users who do not have a REST client installed can still perform API testing through the open source application Swagger UI, which allows users to work with the REST API resources without the need to have any subsequent clients or implementation logic set up through a browser-based test client. Additionally, Swagger UI produces an interactive REST API console for users to provide visual documentation and a built-in method of calling REST services.

4.1 Getting Started

To launch the Swagger UI and start working with the Output Transformation Server REST services, you must open a web browser and navigate to the following URL while replacing the host name and port number with your own configuration information when typing in the address:

http://<host>:<port>/rest

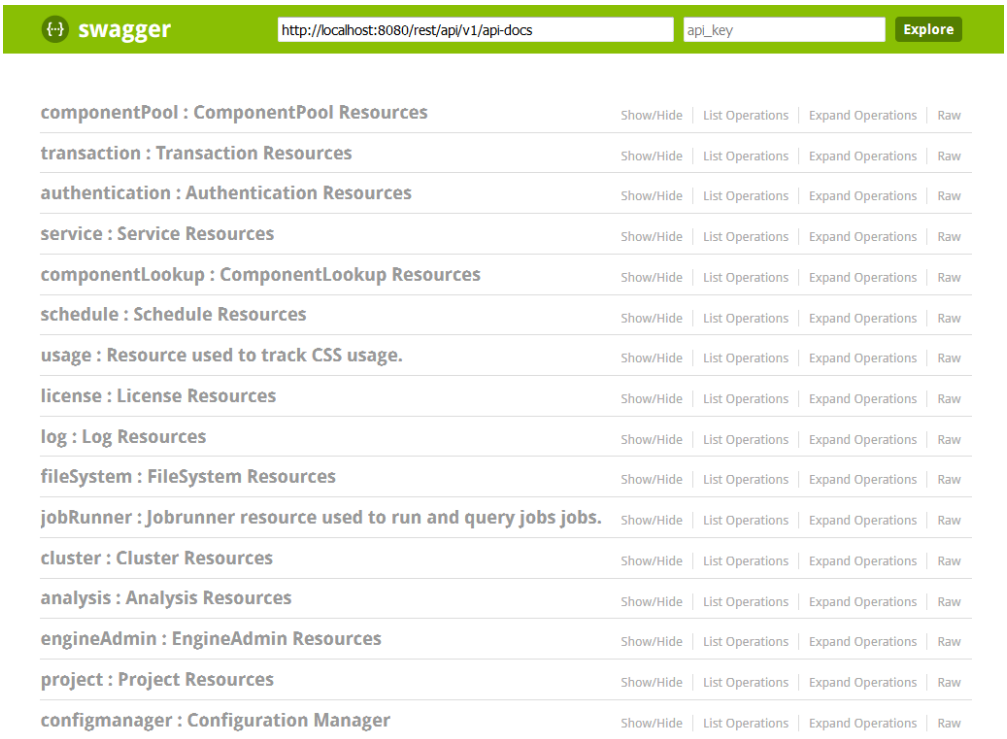

Figure 4-1: Swagger UI main page displaying Output Transformation Server REST services

4.2 Working in Swagger UI

On the main Swagger UI page, a list of all Output Transformation Server REST services is shown. Beside the name of each service, you can collapse or expand the panes displaying its operations by clicking **Show/Hide**. The operations belonging to a service can also be displayed by clicking the **List Operations** option. Furthermore, you can select **Expand Operations** to show specific details about an operation. When an operation is expanded, the following details are displayed:

- **Response Class (Status)**. Response classes can be switched between the response JSON schema using **Model Schema** or the response class model with brief descriptions of each field using **Model**.
- **Parameters**. Input parameters are shown with its description, type information, and a text field for each parameter value.
- **Request Class**. Request classes can be switched between the response JSON schema using **Model Schema** or the response class model with brief descriptions of each field using **Model**.
- **Response Messages**. Listing of possible response messages.

4.2.1 Testing Operations

If you want to create your own client for these APIs, you can test these operations in Swagger UI to see how they work and the expected responses.

To test an operation:

Note: When testing operations, it is recommended you populate all input parameter fields.

1. 1. On the main Swagger UI screen, select the operation you want to test and expand it.
2. 2. In the **Parameters** section, click **Model Schema** to pre-populate the field with the JSON template. This can then be used to create a complete population value.
3. Click **Try it out!**.

   The Response body appears below.

4.3 Calling the JobRunner Sample

Before you can call the JobRunner web service, you must authenticate the user account that will be used to call it by acquiring a token ID.

To set up authentication for the user:

1. On the main Swagger UI screen, click the **authentication** service to expand it.
2. Click the **POST /authentication/login** operation to expand it.
3. Locate the **body** input parameter and in the corresponding **Value** field, type the following:

```
{
     “user”: “JobRunner”,
     “password”: “JobRunner”
}
```

4. Click **Try it out!**.
5. In the **Response Body** field, locate the value of the **tokenID** property and copy it to the clipboard.

   With the tokenID established for the user, you can proceed to set up a call to the JobRunner sample web service.

To call the JobRunner sample web service:

1. On the main Swagger UI screen, click the **jobRunner** service to expand it.
2. Click the **POST /jobRunner/job** operation to expand it.
3. In the **token** parameter’s **Value** field, paste the **tokenID** obtained during the authentication service set up. This value is required.

4. Copy the JSON expression shown in **Appendix A** and paste it into the body parameter's **Value** field. This value is also required.

   Note: The value of the **data** property under **inputMap** is base64 encoded.

5. Click **Try it out!**.

   The results are displayed in the **Response Body** field, which should match the sample shown in **Appendix B**.

   Note: The value of the **data** property under **outputMap** is base64 encoded.

4.3.1 Appendix A

JSON expression for the **POST /jobRunner/job body** input parameter.

```
{
  "jobVariables": {
    "keys": ["jobvar1"],
    "values": [{"value": "value1"}]
  },
  "inputXdocs": {
    "xdocKeys": ["DataToConvert"],
    "multiXDocKeys": [],
    "xdocData": [{"data":
"Q291bnRyeSwiUHJvdmluY2UtVGVycmlOb3J5IE5hbWUiLCJQb3B1bGF0aW9uIChpbiAxMDAwJ3MpIiwiQXJlYSBT
aXplIChpbiBrbTIpIiwiVHlwZSINCkNhbmFkYSwiQWxiZXJOYSIsIjMwOTcuNSIsIjY2MTE4NSIsIlByb3ZpbmNlI
gOKQ2FuYWRhLCJCcmlOaXNoIENvbHVtYmlhIiwiNDA5Mi44IiwiOTQ4NTk2IiwiUHJvdmluY2UiDQpDYW5hZGEsIk
1hbmlOb2JhIiwiMTE1Ni45IiwiNjUwMDg2IiwiUHJvdmluY2UiDQpDYW5hZGEsIk5ldyBCcnVuc3dpY2siLCI3NjA
uNiIsIjczNDM3IiwiUHJvdmluY2UiDQpDYW5hZGEsTmV3Zm91bmRsYW5kLCI1MzQuNCIsIjQwNDUxNyIsIlByb3Zp
bmNlIgOKQ2FuYWRhLCJOb3JOaHdlc3QgVGVycmlOb3JpZXMiLCIOMy4xIiwiMTE3ODI4MyIsIlRlcnJpdG9yeSINC
kNhbmFkYSwiTm92YSBTY29OaWEiLCI5NTIuNCIsIjU1NDkwIiwiUHJvdmluY2UiDQpDYW5hZGEsIk51bmF2dXQiLC
IyOC4OIiwiMjIwMTQwMCIsIlRlcnJpdG9yeSINCkNhbmFkYSxPbnRhcmlvLCIxMTg4OC43IiwiMTA2ODU4MiIsIlB
yb3ZpbmNlIgOKQ2FuYWRhLCJQcmluY2UgRWR3YXJkIElzbGFuZCIsIjEzOS44IiwiNTY1NyIsIlByb3ZpbmNlIgOK
Q2FuYWRhLCJRdWViZWMiLCI3NDM4LjUiLCIxNTQwNjgwIiwiUHJvdmluY2UiDQpDYW5hZGEsIlNhc2thdGNoZXdhb
iIsIjEwMzUuNSIsIjY1MTkwMCIsIlByb3ZpbmNlIgOKQ2FuYWRhLCJZdWtvbiIsIjI5LjMiLCI1MzYzMjQiLCJUZX
JyaXRvcnkiDQpDaGluYSwiQW5odWkiLCI2MTUzMS44IiwiMTM5OTAwIiwiUHJvdmluY2UiDQpDaGluYSxBb21lbiw
ONjguNiwyMSxQcm92aW5jZQOKQ2hpbmEsQmVpamluZywxMzYwOC42LDE3ODAwLFByb3ZpbmNlDQpDaGluYSxDaG9u
Z3FpbmcsMzIxMTkuMCw4MjM2NixQcm92aW5jZQOKQ2hpbmEsRnVqaWFuLDM2NzEwLjEsMTIzMTAwLFByb3ZpbmNlD
QpDaGluYSxHYW5zdSwyNzQONS4wLDM2NjUwMCxQcm92aW5jZQOKQ2hpbmEsR3Vhbmdkb25nLDc1MTk4LjMsMTk3MT
AwLFByb3ZpbmNlDQpDaGluYSwiR3Vhbmd4aSBaaHVhbmciLDUxMjE1LjksMjIwNDAwLFRlcnJpdG9yeQOKQ2hpbmE
sR3VpemhvdSwzODkyNy4zLDE3NDAwMCxQcm92aW5jZQOKQ2hpbmEsSGFpbmFuLDgxMTUuMiwzNDMwMCxQcm92aW5j
ZQOKQ2hpbmEsSGViZWksNzEONjEuMiwyMDI3MDAsUHJvdmluY2UNCkNoaW5hLEhlaWxvbmdqaWFuZywzOTAxOC44L
DQ2MzYwMCxQcm92aW5jZQOKQ2hpbmEsSGVuYW4sMTAyMzE2LjksMTY3MDAwLFByb3ZpbmNlDQpDaGluYSxIdWJlaS
w2NDI1NS4yLDE4NzUwMCxQcm92aW5jZQOKQ2hpbmEsSHVuYW4sNzAzMTcuMSwyMTA1MDAsUHJvdmluY2UNCkNoaW5
hLEppYW5nc3UsNzgxODguMSwxMDIyMDAsUHJvdmluY2UNCkNoaW5hLEppYW5neGksNDUzOTEuOCwxNjQ4MDAsUHJv
dmluY2UNCkNoaW5hLEppbGluLDI4NTIxLjksMTg3MDAwLFByb3ZpbmNlDQpDaGluYSxMaWFvbmluZywwONTQ3NC4yL
DE1MTAwMCxQcm92aW5jZQOKQ2hpbmEsIk5laSBNb25nZ29sIiwyNDQ3MS4wLDExNzc1MDAsVGVycmlOb3J5DQpDaG
luYSwiTmluZ3hpYSBIdWkiLDU5NDYuOCw2NjQwMCxUZXJyaXRvcnkNCkNoaW5hLFFpbmdoYWksNTMyOS44LDcyMTA
wMCxQcm92aW5jZQOKQ2hpbmEsU2hhYW54aSwzODMwOS43LDE5NTgwMCxQcm92aW5jZQOKQ2hpbmEsU2hhbmRvbmcs
OTY2ODMuMCwxNTMzMDAsUHJvdmluY2UNCkNoaW5hLFNoYW5naGFpLDE1OTAyLjAsNTgwMCxQcm92aW5jZQOKQ2hpb
mEsU2hhbnhpLDMzMjk2LjIsMTU3MTAwLFByb3ZpbmNlDQpDaGluYSxTaWNodWFuLDg4OTgxLjQsNDg3MDAwLFByb3
ZpbmNlDQpDaGluYSxUaWFuamluLDEwMzg5LjUsMTEzMDAsUHJvdmluY2UNCkNoaW5hLFhpYW5nZ2FuZyw3MjYwLjk
sMTA5MixUZXJyaXRvcnkNCkNoaW5hLCJYaW5qaWFuZyBVeWd1ciIsMTkxMjcuNSwxNjQ2ODAwLFRlcnJpdG9yeQOK
Q2hpbmEsWGl6YW5nLDI3NTcuNywyMjE2MDAsUHJvdmluY2UNCkNoaW5hLFl1bm5hbiwOMzc5NS41LDEyMjE2MDAsU
HJvdmluY2UNCkNoaW5hLFpoZWppYW5nLDQ1NDcwLjEsMTAxODAwLFByb3ZpbmNlDQo="}],
    "multiXDocData": [{
        "multiXDocKeys": [],
        "multiXDocData": []
      }
    ]
  },
  "outputXdocs": {
    "xdocKeys": [],
```

```
    "multiXDocKeys": [],
    "multiXDocData": [{
        "multiXDocKeys": [],
        "multiXDocData": []
       }
    ]
  },
  "filteredResult": false,
  "runJobNow": false,
  "includedJobVars": [],
  "includedInputDocs": [],
  "includedResultDocs": [],
  "ignoreServerError": false,
  "componentName": "_sample/OutputTransformationServer/DataTransformation/processFlow/
Sample-csv2xml.xProcessFlow"
}
```


4.3.2 Appendix B

JSON expression for the **POST /jobRunner/job body** parameter's response body.


```
{
  "result": {
    "alias": "",
    "component": "_sample/OutputTransformationServer/DataTransformation/processFlow/
Sample-csv2xml.xProcessFlow",
    "errorNumber": 0,
    "failureMessage": "",
    "fatalErrorNumber": 0,
    "inputSize": 2051,
    "invocationMethod": "",
    "jobId": "20180612_1042_001",
    "outputSize": 7352,
    "queuedMs": 0,
    "runningMs": 251,
    "startTime": 1528814536713,
    "status": "FINISHED",
    "stopTime": 1528814536964,
    "suspendMs": 0,
    "totalMs": 251,
    "userField": "",
    "warningNumber": 0,
    "inputMap": {
      "xdocKeys": [
        "DataToConvert"
      ],
      "multiXDocKeys": null,
      "xdocData": [
        {
          "data": "Q291bnRyeSwiUHJvdmluY2UtVGVycmlOb3J5IE5hbWUiLCJQb3B1bGF0aW9uIChpbi…
DQo="
        }
      ],
      "multiXDocData": null
    },
    "resultMap": {
      "xdocKeys": null,
      "multiXDocKeys": [
        "TARGET_1"
      ],
      "xdocData": null,
      "multiXDocData": [
        {
          "multiXDocKeys": [
            "0"
          ],
          "multiXDocData": [
            {
              "data":
```

```
"PD94bWwgdmVyc2lvbjOiMS4wIiBlbmNvZGluZzOiVVRGLTgiPz4NCjxQb3B1bGFOaW9……..cnJpdG9yaWVzPgOKI
CAgIDwvQ291bnRyeT4NCjwvUG9wdWxhdGlvbkRlbnNpdHk+DQo="
            }
          ]
        }
      ]
    },
    "jobVarMap": {
      "keys": [
        "java.vendor",
        "xenos.base.repository",
        "base.repository.dir",
        "jobvar1",
        "JobTicket.id",
         ……... ,
        "ots.mode"
      ],
      "values": [
        {
          "value": "Oracle Corporation",
          "values": null
        },
        {
          "value": "SUN_STANDARD",
          "values": null
        },
        {
          "value": "C:\\OpenText\\OTS_BASE\\TomcatBase\\ots-tc",
          "values": null
        },
          "value": "C:/OpenText/OTS_BASE/BaseRepositories/tomcat/ots-tc",
          "values": null
        },
        {
          "value": "value1",
          "values": null
        },
        {
          "value": " 20180612_1042_001",
          "values": null
        }
      ]
    }
  },
  "message": null,
  "status": "success"
}
```


4.3.3 Appendix C

Sample of Java JobRunner client class.

```
import java.util.*;

import com.xenos.framework.client.jobrunner.*;
import com.xenos.framework.component.*;
import com.xenos.framework.system.jas.JobStats;

public class JobRunnerSample {

    public static void main(String[] args) {
        JobRunner runner =
JobRunnerFactory.constructRestWebServiceJobRunner("localhost", 8080, null);
        String runableCSV = "_sample/OutputTransformationServer/DataTransformation/
processFlow/Sample-csv2xml.xProcessFlow";
        String inputCSV = "Country,\"Province-Territory Name\",\"Population (in 1000's)
\",\"Area Size (in km2)\",\"Type\"\nCanada,\"Alberta\",\"3097.5\",\"661185\",\"Province
\"\nCanada,\"British Columbia\",\"4092.8\",\"948596\",\"Province\"\nCanada,\"Manitoba\",
\"1156.9\",\"650086\",\"Province\"\n";
```

```
        Map inputMap = new HashMap();
        inputMap.put("DataToConvert", new XDoc(inputCSV));
        Map out = new HashMap();
        out.put("TARGET_1", new MultiXDoc());
        Map varMap = new HashMap();

        IJobResult jobResult = runner.runJob(runableCSV, inputMap, out, null, null);
        jobResult.waitForJobToFinish();

        System.out.println("Job result: ");
        for (int i = 0; i < jobResult.getResultCount(); i ++){
            XDoc x = jobResult.getResultFor(i);
            System.out.println(x.toString());
        }

        System.out.println("Job Stats: ");
        JobStats stats = jobResult.getJobStats();
        System.out.println(stats.toString());
        runner.close();

    }
}
```

Chapter 5

Repository Services 2

Repository Services 2 is a OpenText Output Transformation Server service that enables communication between Output Transformation Server, or licensed OpenText products, and OpenText Output Archive and other third-party repositories.

5.1 Repository Services 2 API Interface

Repository Services 2 is a core Output Transformation Server service used to communicate with repository adapters from a common interface in order to access content residing in external repositories.

There are two ways to invoke Repository Services 2:

- Through a Javadoc API
- As a web service

This section focuses on using the Repository Services 2 API interface.

5.1.1 Repository Adapters for Repository Services 2

OpenText currently supports connections to the following repositories:

- Content Management Interoperability Services (CMIS) with the CMISAdapter
- Documentum servers with the DocumentumAdapter
- IBM FileNet P8 with the FileNetP8Adapter
- IBM FileNet Image Services with the IsraAdapter
- IBM Content Manager OnDemand with the ODAdapter

These repository adapters are documented in *OpenText Output Transformation Server User Guide*.

5.1.2 Repository Services 2 Methods

Once the Repository Adapter has been configured and started, Output Transformation Server can access documents in the repositories to use in process flows, document transformations, or document presentment.

To enable these tasks, Repository Services 2 has the following methods which are used in Javadocs:

- **advancedSearch**. Performs an advanced search.
- **findApplications**. Retrieves a list of available applications from a repository adapter based on some filter expression.
- **getAboutInfo**. Returns information about Repository Services 2.
- **getCustomTree**. Returns a user-defined tree view from a repository adapter.
- **getPushNotifications**. Enables request that is meant to block until some "real-time" notification is available.
- **getSupportedOptions**. Returns a list of supported options from a repository for each user.
- **getTreeView**. Returns a tree view from a repository adapter based on some combination of path and filter.
- **getViewAsOptions**. Returns all possible transform options.
- **invokeMethod**. Dynamically invokes a method on the adapter. For example, it can be used to invoke **searchFiles** to search for files in the repository, **deleteFiles** to delete specified files, or **deleteDocument** to mark a specified document as invisible.
- **loadDocument**. Loads documents into the repository.
- **login**. Logs into a repository adapter.
- **logout**. Logs out of a repository adapter.
- **ping**. Sends an object to the Repository Service and returns that object.
- **pingAdapter**. Pings the adapter.
- **readToc**. Returns the table of contents for a specific application from a repository adapter.
- **retrieveDocument**. Retrieves the specified document from a repository adapter.
- **retrieveDocumentMetaData**. Retrieves metadata properties and/or resource bytes based on the DocumentMetaOptions.
- **retrieveDocumentWithOptions**. Retrieves the specified document from a repository adapter, and may perform additional post-processing based on the retrieve options.
- **retrieveFields**. Retrieves searchable fields associated with the given request.

- **retrieveHitList**. Retrieves a list of hits that correspond to the given search criteria.
- **retrieveInstances**. Retrieves a list of running repository adapter instances.
- **setCustomTree**. Sends a user-defined tree view to a repository adapter.

For more information, refer to the Repository Services 2 Javadocs.

5.1.3 RepositoryAdapterSampleClientV2

The **RepositoryAdapterSampleClientV2** connects to OpenText Repository Services 2 directly using API. When you run the process, it calls the methods to query adapter information, retrieves document information, and loads the document. The results appear in the log in the Output window.

This sample client and code can be used to test your existing adapter setup or to create your own client for accessing your repositories through the Repository Services.

To view sample code for the RepositoryAdapterSampleClientV2, navigate to `<install_home>\install\<version>\initialFiles\common\com\xenos\framework\process\repository\v2` and double-click **RepositoryAdapterSampleClientV2.java**. The results appear in the log in the Output window. You can also view both sample code and sample output in the related links.

If you add a RepositoryAdapterSampleClientV2 component to a process flow or an event, ensure the component maps to existing documents or the process will fail.

The parameters for the RepositoryAdapterSampleClientV2 component are:

Connection	Provides configuration properties for connecting to OpenText Repository Services 2.
ConnectionType	Determines whether to connect directly (in-process) or via web service. If **DIRECT** is selected, the **ServiceConfig** value will be used. If **WEB_SERVICE** is selected, the **ServiceUrl** value will be used.
ServiceConfig	Specifies the Repository Adapter Service configuration file, which is used by the Repository Adapter to connect directly to Output Transformation Server via API.
ServiceUrl	Designates the Repository Services 2 web service URL.
User	Specifies the user name for the Output Archive login.
Password	Specifies the password that corresponds to the above **User** property.

Instance	Indicates the name of the instance to use.
Application	Indicates the application to search or retrieve. The format of this value will vary depending on what adapter type you are using. For the IBM ODAdapter (OnDemand), the format is `<Folder>`.
DocName	Identifies the name or ID of the document to retrieve.
SearchCriteria	Defines the name and value pairs that represent a query and returns values in an array from the method called, such as retrieveHitList.
AutoRetrieveOnSingleHit	Sets an option for the **SearchCriteria** property to automatically retrieve a document if a search returns only one hit.

Note: The **RepositoryAdapterSampleClientV2.java** code, which calls the methods, needs a parameter definition (.parmdef) and component definition to further support it. The parameter definition and component definition for this component are located in the same place as the sample code.

Related Links

- "Repository Services 2 Methods" on page 66
- "RepositoryAdapterSampleClientV2 Sample Java Class" on page 68
- "RepositoryAdapterSampleClientV2 Sample Output" on page 80
- *OpenText Output Transformation Designer - User Guide (VDTOTS-H-UTD)*

5.1.3.1 RepositoryAdapterSampleClientV2 Sample Java Class

```
package com.xenos.framework.process.repository.v2;
import java.io.*;
import com.xenos.d2e.parm.*;
import com.xenos.framework.*;
import com.xenos.framework.bpengine.*;
import com.xenos.framework.bpengine.parm.*;
import com.xenos.framework.component.*;
import com.xenos.framework.exceptions.*;
import com.xenos.framework.process.repository.v2.parm.*;
import com.xenos.framework.util.message.*;
import com.xenos.repository.v2.api.*;
import com.xenos.repository.v2.client.*;
```

```
 /**
  * A code sample demonstrating how to use <code>RepositoryClientHelper</code>
  * or <code>RepositoryWebServiceClientHelper</code> to interact with Xenos
  * Repository Services.  This code is not intended for production use.
  */
 public class RepositoryAdapterSampleClientV2 extends BaseComponent implements IProcess {
 /* A list of the required job variables this process will use. */
     private static JobVarDef JOB_VAR[] = JobVarDef.EMPTY_ARRAY;

     /* A list of required source XDocs this process will use. */
     private static SourceResultDef REQUIRED_SOURCES[] = SourceResultDef.EMPTY_ARRAY;

     /* A list of result XDocs this process will create. */
     private static SourceResultDef CREATED_RESULTS[] = new SourceResultDef[] {
     new SourceResultDef("document", "The document retrieved from the repository
adapter."),
     new SourceResultDef("resources", "The document resources associated with the
retrieved document."),
     new SourceResultDef("hitList", "The hit list in XML format.")
     };

     /* Routing for this process. */
     private static String STATIC_FLOWS[] = new String[] { "next" };

     /* Contains the Component's configuration. */
     private RepositoryAdapterSampleClientV2Parm m_parms;

 private IRepositoryServiceV2 m_client;
     /**
      * Called to pass in the components parameters.  This is called
      * once before initialize() is called.
      * @param po the parm handler for this component.
      */
     public void configureParameters(ParmHandler po) {
         m_parms = (RepositoryAdapterSampleClientV2Parm)po;
         super.configureParameters(po);
```

```
    }
    /**
     * Called once before this component is used (run, started or otherwise
     * used by other components.
     */
    public void initialize() {
        IXenosLogger logger = getLogger();
        if (logger.isDebugEnabled()) {
            logger.log("initializing...");
        }
    }
    /**
     * Called when the component runs.
     */
    public void runComponent() {
        // Processes are runable; just call performProcess
        performProcess();
    }
    /**
     * Causes the work for this process to be done, the return
     * value is the output number (index) which the flow should
     * move to.
     */
    public int performProcess() {
        IXenosLogger logger = getLogger();
        if (logger.isDebugEnabled()) {
            logger.log("parms in use...\n" + m_parms.toReportString(0));
        }
        JobTicket jt = getJobTicket();
    try {
    RepositoryAdapterConnectionParm conn = m_parms.getConnection();
    if (conn.getConnectionType() == ConnectionTypeEnum.DIRECT) {
    String adapterConfig = conn.getServiceConfig();
    // start service if it hasn't already been started
    getSystem().obtainService(adapterConfig);
```

```
    m_client = new RepositoryClientHelper();
    } else {
    String url = conn.getServiceUrl();
    m_client = new RepositoryWebServiceClientHelper(url);
    }

    // login
    RepositorySession session = login(conn);

    // show adapter info
    listAdapterInfo(session);

    // retrieve hit list
    HitList hitList = retrieveHitList(session);
    if (hitList != null) {
    XDoc hitListDoc = hitListToXml(hitList);
    jt.putResultMap(CREATED_RESULTS[2].getName(), hitListDoc);
    }

    // check if document was automatically retrieved from hit list
    DocumentResult document = hitList.getDocument();

    // retrieve documents and resources if hitlist did not contain a document already
    if (document == null) {
    document = retrieveDocument(session, m_parms.getApplication(),
m_parms.getDocName());
    }
    // assign to xdoc
    if (document != null) {
    byte[] bytes = document.getDocBytes();
            XDoc xdoc = new XDoc(bytes);
            int docLength = bytes.length;
            jt.putResultMap(CREATED_RESULTS[0].getName(), xdoc);

            byte[] resourcesBytes = document.getResourceData();
```

```
            if (resourcesBytes != null) {
            XDoc resources = new XDoc(resourcesBytes);
            jt.putResultMap(CREATED_RESULTS[1].getName(), resources);
            }
    }

    logout(session);
    } catch (Exception ex) {
            throw new GenericRuntimeException("RepositoryServicesSampleClientV2 Process
Failed", ex);
    }

        return 0; // pick one of the static or dynamic flows
    }
    private RepositorySession login(RepositoryAdapterConnectionParm conn) throws
Exception {

    RepositoryCredentials credentials = new RepositoryCredentials();
    credentials.setLogin(conn.getUser());
    credentials.setPassword(conn.getPassword());
    credentials.setInstanceName(conn.getInstance());

    RepositoryLoginResult loginResult = m_client.login(credentials);

        GenericMessages.issueDebug(m_logger, "Logged into instance: " +
conn.getInstance() + ".");
    return loginResult.getSession();
    }

    private RepositoryResult logout(RepositorySession session) throws Exception {

    RepositoryResult result = m_client.logout(session);
    return result;
    }

    private void listAdapterInfo(RepositorySession session) throws Exception {
```

```
     InstancesResult instancesResult = m_client.retrieveInstances();
         GenericMessages.issueDebug(m_logger, "Adapter info:");
     RepositoryInstance[] instances = instancesResult.getInstances();
     int size = instances.length;
     for (int x = 0; x < size; x++) {
     RepositoryInstance instance = instances[x];
             GenericMessages.issueDebug(m_logger, "Adapter Name: " + instance.getName());
             GenericMessages.issueDebug(m_logger, " Description: " +
instance.getDescription());
             GenericMessages.issueDebug(m_logger, "        Type: " + instance.getType());
             GenericMessages.issueDebug(m_logger, "\n");
     }
     }

     private HitList retrieveHitList(RepositorySession session) throws Exception {

     int searchCritCount = m_parms.getSearchCriteriaCount();
     if (searchCritCount == 0) {
         GenericMessages.issueDebug(m_logger, "retrieveHitList skipped because no
searchCriteria was specified");
     return null;
     }

 boolean autoRetrieve = m_parms.getAutoRetrieveOnSingleHit();
 SearchCriteria searchCriteria = new SearchCriteria();
 for (int x = 0; x < searchCritCount; x++) {
 SearchCriteriaParm sc = m_parms.getSearchCriteria(x);
 Operator op = resolveOperator(sc.getOperator().toString());
 if (sc.getValue2().trim().length() == 0) {
 searchCriteria.addCriteria(sc.getField(), op, sc.getValue());
 } else {
 searchCriteria.addCriteria(sc.getField(), op, sc.getValue(), sc.getValue2());
 }
 }
```

```
searchCriteria.setApplicationIds(new String[] { m_parms.getApplication() });

RepositoryPropMap options = new RepositoryPropMap();
if (autoRetrieve) {
options.setValue(SearchCriteria.AUTO_RETRIEVE_ON_SINGLE_HIT, Boolean.TRUE.toString());
}
options.setValue(SearchCriteria.USE_APP_ID, Boolean.FALSE.toString());
searchCriteria.setOptions(options);
searchCriteria.setStart(0);
searchCriteria.setLimit(Integer.MAX_VALUE);
searchCriteria.setInterval(Integer.MAX_VALUE);

HitList hitList = m_client.retrieveHitList(session, searchCriteria);

int hitCount = -1;
HitItem[] items = hitList.getAllHits();
if (items != null) {
hitCount = items.length;
for (int x = 0; x < hitCount; x++) {
HitItem item = items[x];
        GenericMessages.issueDebug(m_logger, "Hit " + x + ": " + item.toString());

}
}

        GenericMessages.issueDebug(m_logger, "retrieveHitList called and returned " +
hitCount + " items.");
return hitList;
    }

    private DocumentResult retrieveDocument(RepositorySession session, String appId,
String docId) throws Exception {

    if (m_parms.getDocName().trim().length() == 0) {
        GenericMessages.issueDebug(m_logger, "retrieveDocucment was skipped because no
DocName was specified");
        return null;
```

```
    }

    Document document = new Document();
    document.setApplicationId(appId);
    if (m_parms.getConnection().getConnectionType() == ConnectionTypeEnum.WEB_SERVICE) {
    QuotedString qs = new QuotedString();
    document.setDocId(qs.toQuotedString(docId));
    } else {
        document.setDocId(docId);
    }

    DocumentResult result = m_client.retrieveDocument(session, new Document[]
{ document });
    GenericMessages.issueDebug(m_logger, "Document retrieved: " +
result.getDocBytes().length + " bytes");
    if (result.getResourceData() != null) {
        GenericMessages.issueDebug(m_logger, "Document resources: " +
result.getResourceData().length + " bytes");
     } else {
        GenericMessages.issueDebug(m_logger, "No Document resources,");
     }

    return result;
    }

    private XDoc hitListToXml(HitList hitList) {
        XDoc result = new XDoc();

        try {
        BufferedWriter writer = new BufferedWriter(result.obtainWriter());
        writer.write("<?xml version=\"1.0\" encoding=\"UTF-8\" ?>\n");
        writer.write("<HitList>\n");

        String[] attribNames = hitList.getAttribNames();
        HitItem[] items = hitList.getAllHits();
        for (int x = 0; x < items.length; x++) {
```

```
        HitItem item = items[x];
        String[] attribValues = item.getAttribValues();
            writer.write("    <HitItem>\n");

        writer.write("        <DocumentID>");
            writer.write(item.getDocId().trim());
        writer.write("</DocumentID>\n");
        for (int y = 0; y < attribValues.length; y++) {
        String attrib = attribNames[y];
        attrib = attrib.replace(' ', '_');
        attrib = attrib.replace('$', '_');
        attrib = attrib.trim();
        String value = attribValues[y];
        writer.write("        <" + attrib + ">");
        if (value != null) {
        writer.write(value.trim());
        } else {
        writer.write("");
        }
            writer.write("</" + attrib + ">\n");
        }
            writer.write("    </HitItem>\n");
        }

        writer.write("</HitList>\n");
        writer.close();
        } catch (IOException ioe) {
        ioe.printStackTrace();
        }

        return result;
    }

    private Operator resolveOperator(String operator) {
        Operator op = null;
```

```
        operator = "OP_" + operator;
        if (operator.equals(SearchCriteria.OP_BETWEEN.getName())) {
            op = SearchCriteria.OP_BETWEEN;
        }
        else if (operator.equals(SearchCriteria.OP_EQUAL.getName())) {
            op = SearchCriteria.OP_EQUAL;
        }
        else if (operator.equals(SearchCriteria.OP_GREATER_THAN.getName())) {
            op = SearchCriteria.OP_GREATER_THAN;
        }
        else if (operator.equals(SearchCriteria.OP_GREATER_THAN_EQUAL.getName())) {
            op = SearchCriteria.OP_GREATER_THAN_EQUAL;
        }
        else if (operator.equals(SearchCriteria.OP_IN.getName())) {
            op = SearchCriteria.OP_IN;
        }
        else if (operator.equals(SearchCriteria.OP_LESS_THAN.getName())) {
            op = SearchCriteria.OP_LESS_THAN;
        }
        else if (operator.equals(SearchCriteria.OP_LESS_THAN_EQUAL.getName())) {
            op = SearchCriteria.OP_LESS_THAN_EQUAL;
        }
        else if (operator.equals(SearchCriteria.OP_LIKE.getName())) {
            op = SearchCriteria.OP_LIKE;
        }
        else if (operator.equals(SearchCriteria.OP_NOT_BETWEEN.getName())) {
            op = SearchCriteria.OP_NOT_BETWEEN;
        }
        else if (operator.equals(SearchCriteria.OP_NOT_EQUAL.getName())) {
            op = SearchCriteria.OP_NOT_EQUAL;
        }
        else if (operator.equals(SearchCriteria.OP_NOT_IN.getName())) {
            op = SearchCriteria.OP_NOT_IN;
        }
        else if (operator.equals(SearchCriteria.OP_NOT_LIKE.getName())) {
```

```
            op = SearchCriteria.OP_NOT_LIKE;
        }

        return op;
    }

    /**
     * Analyzes the result of a Xenos Repository Services method invocation.
     *
     * @param result
     * @throws Exception
     */
    private void analyzeResult(RepositoryResult result) throws Exception {
    if (result != null) {


    } else {

    }
    }

    /**
     * @return An Array describing any input sources read by this component.
     */
    public SourceResultDef[] getRequiredSources() {
        // by default, this returns the sources as defined
        // by the component wizard.  If the sources used is
        // based on some configuration, this should be dyanmically
        // created.
        return REQUIRED_SOURCES;
    }
    /**
     * @return An Array describing any job variables used/created by this component.
     */
    public JobVarDef[] getRequiredJobVars() {
```

```
        // by default, this returns the job variables as defined
        // by the component wizard.  If the job variables used is
        // based on some configuration, this should be dyanmically
        // created.
        return JOB_VAR;
    }
    /**
     * @return An Array describing any results created by this component.
     */
    public SourceResultDef[] getCreatedResults() {
        // by default, this returns the results as defined
        // by the component wizard.  If the results is
        // based on some configuration, this should be dyanmically
        // created.
        return CREATED_RESULTS;
    }
    /**
     * Returns the number of routes for this process.
     */
    public int getNumberOfOutput() {
        return 1;
    }
    /**
     *
     */
    public String getOutputName(int index) {
        return STATIC_FLOWS[index];
    }
    public boolean canCreateNewOutput() {
        return false;
    }
    public boolean removeOutput(int index) {
        return false;
    }
    public boolean canRemoveOutput(int index) {
```

```
        return false;
    }
    public BPConditionParm createNewOutput() {
        // should never be called
        return null;
    }
    public BPConditionParm getOutputCondition(int index) {
        // should never be called
        return null;
    }
}
```

5.1.3.2 RepositoryAdapterSampleClientV2 Sample Output

```
2014-08-18T13:11:59.842-04:00 <JobWorker-40> DEBUG : parms in use...
Connection...
    ConnectionType = DIRECT
    ServiceConfig = _sample/ESRE/local.esreAdapter
    ServiceUrl =
    User = Administrator
    Password= ?-?-?
    Instance = local
Application =
DocName =
SearchCriteria[0]...
    Field = fullName
    Operator = LIKE
    Value = %
    Value2 =
AutoRetrieveOnSingleHit = false
2014-08-18T13:12:00.492-04:00 <JobWorker-40> DEBUG : Logged into instance: local.
2014-08-18T13:12:00.493-04:00 <JobWorker-40> DEBUG : Adapter info:
2014-08-18T13:12:00.493-04:00 <JobWorker-40> DEBUG : Adapter Name: local
2014-08-18T13:12:00.494-04:00 <JobWorker-40> DEBUG :  Description: Sample ES Repository
Adapter created by Output Transformation Designer.
2014-08-18T13:12:00.494-04:00 <JobWorker-40> DEBUG :         Type: ESRE Adapter
2014-08-18T13:12:00.495-04:00 <JobWorker-40> DEBUG :
```

```
 2014-08-18T13:12:01.575-04:00 <JobWorker-40> DEBUG : Hit 0: SFWealthStatements/
Stmt_N2_20140711_1311_002.pdf[NFggKCpFQ1JlPlhqdiF-
MDI8SnJwIH1yJVpUZmx2f3t9ZCRBO1lucSMyLX9HO1VNXFtxfiQ2], ext=pdf, vals=[20140711131234,
4555-9999-0001, Sam Shi, 8257.52, 248, USD, 2013-04-21, 2013-03-01, 2013-03-31, 1,
2013-03, 63, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.576-04:00 <JobWorker-40> DEBUG : Hit 1: SFWealthStatements/
Stmt_N2_20140711_1311_002.pdf[Q2cvNzlUUmFOU21_JTAtPOFLWSF_LzMnNGljdXslLiosczNWUGh9IDJBPC5
WSmRccXAgLTNF], ext=pdf, vals=[20140711131234, 4555-9999-0002, Bob Baker, 18435.2, 554,
USD, 2013-04-21, 2013-03-01, 2013-03-31, 2, 2013-03, 63, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.576-04:00 <JobWorker-40> DEBUG : Hit 2: SFWealthStatements/
Stmt_N2_20140711_1311_002.pdf[Jkpyenw3NURXMEpcYm1wIiQuPGRicnFkcUZGWF5ocW1vVnYzLUVaY3Ukf3E
5LUc_Tk1danYo], ext=pdf, vals=[20140711131234, 4555-9999-0003, Pete Powers, 7360.66,
221, USD, 2013-04-21, 2013-03-01, 2013-03-31, 3, 2013-03, 63, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.577-04:00 <JobWorker-40> DEBUG : Hit 3: SFWealthStatements/
Stmt_N2_20140711_1311_002.pdf[UXU9RUdiYHUoYXstMz47TU9ZZy8zQOM1QndxIykzPDg6JOdkXnYrLkBPSjx
kWHhwf34uOOFT], ext=pdf, vals=[20140711131234, 4555-9999-0004, Abigail Adams, 7538.91,
227, USD, 2013-04-21, 2013-03-01, 2013-03-31, 4, 2013-03, 63, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.577-04:00 <JobWorker-40> DEBUG : Hit 4: SFWealthStatements/
Stmt_N2_20140711_1311_002.pdf[Sm42PkBbWWghWnQmLDcORkhSYCgmPDOuO3BqfCIsNTEzekBdV28kJzlIQzV
dUWtpeHcnNDpM], ext=pdf, vals=[20140711131234, 4555-9999-0005, Charlie Jones, 3895.77,
117, GBP, 2013-04-21, 2013-03-01, 2013-03-31, 5, 2013-03, 63, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.578-04:00 <JobWorker-40> DEBUG : Hit 5: SFWealthStatements/
Stmt_N2_20140711_1311_002.pdf[LVF5ISM-
PEteN1FjaXp3KSs1Q2tpeXtreFNNX2VveHR2XXO6NExnanwrJnhANE5GVVRkd3Ov], ext=pdf,
vals=[20140711131234, 4555-9999-0006, Isabelle Scholes, 3213.68, 97, GBP, 2013-04-21,
2013-03-01, 2013-03-31, 6, 2013-03, 63, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.578-04:00 <JobWorker-40> DEBUG : Hit 6: SFWealthStatements/
Stmt_N2_20140711_1311_002.pdf[PGAoMDJNS1ptRmZ4fikmODpEUnp4KCsgLWJcbnR-
JyMlbCxJSWF2eSs6NSdPQ11VZGl5Jiw-], ext=pdf, vals=[20140711131234, 4555-9999-0007, Rachel
Lee, 4207.9, 127, USD, 2013-04-21, 2013-03-01, 2013-03-31, 7, 2013-03, 63, 8],
ret=[null] resId=[null]

 2014-08-18T13:12:01.578-04:00 <JobWorker-40> DEBUG : Hit 7: SFWealthStatements/
Stmt_N2_20140711_1311_002.pdf[O18nLzFMSllsRV93fSglNzlDUXl3Jyt_LGFbbXN9JiIkaytISGB1eCo5NCZ
OQlxUY2h4JSs9], ext=pdf, vals=[20140711131234, 4555-9999-0008, Whitney Wong, 19261, 578,
HKD, 2013-04-21, 2013-03-01, 2013-03-31, 8, 2013-03, 63, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.579-04:00 <JobWorker-40> DEBUG : Hit 8: SFWealthStatements/
Stmt_N2_20140711_1313_001.pdf[NVkhKStGRFNmP1lxdyJ_MTM9S3NyIX5zJltVZ213IHx-
ZSVCPFpvciQzLiBIPFZOXVxyfyU3], ext=pdf, vals=[20140711131321, 4555-9999-0001, Sam Shi,
8341.84, 251, USD, 2013-02-21, 2013-01-01, 2013-01-31, 1, 2013-01, 61, 8], ret=[null]
resId=[null]

 2014-08-18T13:12:01.579-04:00 <JobWorker-40> DEBUG : Hit 9: SFWealthStatements/
Stmt_N2_20140711_1313_001.pdf[QmYuNjhTUWBzTGx-
JC8sPkBKWCB_LjImM2hidHokLSkrcjJVT2d8fzFAOy1VSWNbcG9_LDJE], ext=pdf,
vals=[20140711131321, 4555-9999-0002, Bob Baker, 18169.4, 546, USD, 2013-02-21,
2013-01-01, 2013-01-31, 2, 2013-01, 61, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.580-04:00 <JobWorker-40> DEBUG : Hit 10: SFWealthStatements/
Stmt_N2_20140711_1313_001.pdf[JOtze3O4NkVYMUtdY3RxIyUvPWVkc3JlckdHWV9pcm5wV3cOLkZbZHYlIHI
6LkhATO5ea3cp], ext=pdf, vals=[20140711131321, 4555-9999-0003, Pete Powers, 7234.82,
218, USD, 2013-02-21, 2013-01-01, 2013-01-31, 3, 2013-01, 61, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.580-04:00 <JobWorker-40> DEBUG : Hit 11: SFWealthStatements/
Stmt_N2_20140711_1313_001.pdf[UHQ8REZhX24nYHosMjO6TE5YZi4zQkIOQXZwIigyOzc5JkZjXXUqLT9OSTt
jV3dvfnOtOkBS], ext=pdf, vals=[20140711131321, 4555-9999-0004, Abigail Adams, 7354.98,
221, USD, 2013-02-21, 2013-01-01, 2013-01-31, 4, 2013-01, 61, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.581-04:00 <JobWorker-40> DEBUG : Hit 12: SFWealthStatements/
Stmt_N2_20140711_1313_001.pdf[JOtze3O4NkVYMUtdY3RxIyUvPWVkc3RlckdHWV9pcm5wV3cOLkZbZHYlIHI
6LkhATO5ea3cp], ext=pdf, vals=[20140711131321, 4555-9999-0005, Charlie Jones, 3961.52,
119, GBP, 2013-02-21, 2013-01-01, 2013-01-31, 5, 2013-01, 61, 8], ret=[null] resId=[null]
```

```
 2014-08-18T13:12:01.581-04:00 <JobWorker-40> DEBUG : Hit 13: SFWealthStatements/
Stmt_N2_20140711_1313_001.pdf[QmYuNjhTUWBzTGx-
JC8sPkBKWCB_LjYmM2hidHokLSkrcjJVT2d8fzFAOy1VSWNbcG9_LDJE], ext=pdf,
vals=[20140711131321, 4555-9999-0006, Isabelle Scholes, 2603.22, 79, GBP, 2013-02-21,
2013-01-01, 2013-01-31, 6, 2013-01, 61, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.581-04:00 <JobWorker-40> DEBUG : Hit 14: SFWealthStatements/
Stmt_N2_20140711_1313_001.pdf[NVkhKStGRFNmP1lxdyJ_MTM9S3NyISRzJltVZ213IHx-
ZSVCPFpvciQzLiBIPFZOXVxyfyU3], ext=pdf, vals=[20140711131321, 4555-9999-0007, Rachel
Lee, 3210.5, 97, USD, 2013-02-21, 2013-01-01, 2013-01-31, 7, 2013-01, 61, 8], ret=[null]
resId=[null]

 2014-08-18T13:12:01.582-04:00 <JobWorker-40> DEBUG : Hit 15: SFWealthStatements/
Stmt_N2_20140711_1313_001.pdf[NFggKCpFQ1JlPlhqdiF-
MDI8SnJxICRyJVpUZmx2f3t9ZCRBO1lucSMyLX9HO1VNXFtxfiQ2], ext=pdf, vals=[20140711131321,
4555-9999-0008, Whitney Wong, 24792.6, 744, HKD, 2013-02-21, 2013-01-01, 2013-01-31, 8,
2013-01, 61, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.582-04:00 <JobWorker-40> DEBUG : Hit 16: SFWealthStatements/
Stmt_N2_20140711_1313_002.pdf[PGAoMDJNS1ptRmZ4fikmODpEUnp6KCUgLWJcbnR-
JyMlbCxJSWF2eSs6NSdPQ11VZGl5Jiw-], ext=pdf, vals=[20140711131346, 4555-9999-0001, Sam
Shi, 8341.97, 251, USD, 2013-03-21, 2013-02-01, 2013-02-28, 1, 2013-02, 62, 8],
ret=[null] resId=[null]

 2014-08-18T13:12:01.583-04:00 <JobWorker-40> DEBUG : Hit 17: SFWealthStatements/
Stmt_N2_20140711_1313_002.pdf[LVF5ISM-
PEteN1FjaXp3KSs1Q2treXdreFNNX2VveHR2XXO6NExnanwrJnhANE5GVVRkd3Ov], ext=pdf,
vals=[20140711131346, 4555-9999-0002, Bob Baker, 18158.9, 545, USD, 2013-03-21,
2013-02-01, 2013-02-28, 2, 2013-02, 62, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.583-04:00 <JobWorker-40> DEBUG : Hit 18: SFWealthStatements/
Stmt_N2_20140711_1313_002.pdf[Sm42PkBbWWghWnQmLDcORkhSYCgoPDsuO3BqfCIsNTEzekBdV28kJzlIQzV
dUWtpeHcnNDpM], ext=pdf, vals=[20140711131346, 4555-9999-0003, Pete Powers, 7254.37,
218, USD, 2013-03-21, 2013-02-01, 2013-02-28, 3, 2013-02, 62, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.584-04:00 <JobWorker-40> DEBUG : Hit 19: SFWealthStatements/
Stmt_N2_20140711_1313_002.pdf[UXU9RUdiYHUoYXstMz47TU9ZZy81QOM1QndxIykzPDg6JOdkXnYrLkBPSjx
kWHhwf34uOOFT], ext=pdf, vals=[20140711131346, 4555-9999-0004, Abigail Adams, 7450.7,
224, USD, 2013-03-21, 2013-02-01, 2013-02-28, 4, 2013-02, 62, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.584-04:00 <JobWorker-40> DEBUG : Hit 20: SFWealthStatements/
Stmt_N2_20140711_1313_002.pdf[Jkpyenw3NURXMEpcYm1wIiQuPGRkcnNkcUZGWF5ocW1vVnYzLUVaY3Ukf3E
5LUc_Tk1danYo], ext=pdf, vals=[20140711131346, 4555-9999-0005, Charlie Jones, 3555.45,
107, GBP, 2013-03-21, 2013-02-01, 2013-02-28, 5, 2013-02, 62, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.585-04:00 <JobWorker-40> DEBUG : Hit 21: SFWealthStatements/
Stmt_N2_20140711_1313_002.pdf[Q2cvNzlUUmF0U21_JTAtP0FLWSEhLzcnNGljdXslLiosczNWUGh9IDJBPC5
WSmRccXAgLTNF], ext=pdf, vals=[20140711131346, 4555-9999-0006, Isabelle Scholes, 2642.6,
80, GBP, 2013-03-21, 2013-02-01, 2013-02-28, 6, 2013-02, 62, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.585-04:00 <JobWorker-40> DEBUG : Hit 22: SFWealthStatements/
Stmt_N2_20140711_1313_002.pdf[NFggKCpFQ1JlPlhqdiF-
MDI8SnJyICNyJVpUZmx2f3t9ZCRBO1lucSMyLX9HO1VNXFtxfiQ2], ext=pdf, vals=[20140711131346,
4555-9999-0007, Rachel Lee, 3093.6, 93, USD, 2013-03-21, 2013-02-01, 2013-02-28, 7,
2013-02, 62, 8], ret=[null] resId=[null]

 2014-08-18T13:12:01.585-04:00 <JobWorker-40> DEBUG : Hit 23: SFWealthStatements/
Stmt_N2_20140711_1313_002.pdf[NVkhKStGRFNmP1lxdyJ_MTM9S3NzISVzJltVZ213IHx-
ZSVCPFpvciQzLiBIPFZOXVxyfyU3], ext=pdf, vals=[20140711131346, 4555-9999-0008, Whitney
Wong, 14885.6, 447, HKD, 2013-03-21, 2013-02-01, 2013-02-28, 8, 2013-02, 62, 8],
ret=[null] resId=[null]

 2014-08-18T13:12:01.586-04:00 <JobWorker-40> DEBUG : retrieveHitList called and
returned 24 items.

 2014-08-18T13:12:01.588-04:00 <JobWorker-40> DEBUG : retrieveDocucment was skipped
because no DocName was specified
```

5.2 Repository Services 2 Web Service Interface

Repository Services 2 is a core Output Transformation Server service used to communicate with repository adapters from a common interface in order to access content residing in external repositories.

There are 2 ways to invoke Repository Services 2:

- Through a Javadoc API
- As a web service

This section focuses on using the Repository Services 2 Web Service interface.

Repository Services 2 Web Service Interface Methods

For a complete list of the Repository Services 2 Web Service methods, please see"Repository Services 2 Methods" on page 66, or refer to the Javadocs, which can be found here:

```
<install_home>/docs/help/mergedProjects/Output Transformation Server
Javadoc/reposervices2
```

5.2.1 Repository Services 2 WSDL

The WSDL document for the Repository Services 2 web service interface is located here:

```
http://<server>:<port>/axis2/services/RepositoryWebService?wsdl
```

The contents of the WSDL document are as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://
org.apache.axis2/xsd" xmlns:ns="http://ws.v2.repository.xenos.com" xmlns:ax213="http://
api.v2.repository.xenos.com/xsd" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/
XMLSchema" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://
schemas.xmlsoap.org/wsdl/soap/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://ws.v2.repository.xenos.com">
```

```
    <wsdl:documentation>RepositoryWebService</wsdl:documentation>
```

```
    <wsdl:types>
```

```
        <xs:schema xmlns:ax214="http://api.v2.repository.xenos.com/xsd"
attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://
ws.v2.repository.xenos.com">
```

```
            <xs:import namespace="http://api.v2.repository.xenos.com/xsd"/>
```

```
            <xs:element name="RepositoryWebServiceException">
```

```
                <xs:complexType>
```

```
                    <xs:sequence>
```

```
                        <xs:element minOccurs="0" name="RepositoryWebServiceException"
nillable="true" type="ns:Exception"/>
```

```
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:complexType name="Exception">
                <xs:sequence>
                    <xs:element minOccurs="0" name="Message" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:element name="setCustomTree">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="tree" nillable="true"
type="ax214:TreeRequest"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="setCustomTreeResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:RepositoryResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveInstances">
                <xs:complexType>
                    <xs:sequence/>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveInstancesResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:InstancesResult"/>
                    </xs:sequence>
```

```
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveHitList">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="criteria" nillable="true"
type="ax214:SearchCriteria"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveHitListResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:HitList"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveFields">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="request" nillable="true"
type="ax214:FieldQueryRequest"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveFieldsResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:FieldQueryResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
```

```
            <xs:element name="retrieveDocumentWithOptions">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="docs"
nillable="true" type="ax214:Document"/>
                        <xs:element minOccurs="0" name="options" nillable="true"
type="ax214:RetrieveOptions"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveDocumentWithOptionsResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:DocumentResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveDocumentMetaData">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="doc" nillable="true"
type="ax214:Document"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveDocumentMetaDataResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:DocumentMetaResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveDocument">
```

```
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="docs"
nillable="true" type="ax214:Document"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="retrieveDocumentResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:DocumentResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="readToc">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="tocRequest" nillable="true"
type="ax214:TocRequest"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="readTocResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:TocResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="pingAdapter">
                <xs:complexType>
                    <xs:sequence>
```

```
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="pong" nillable="true"
type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="pingAdapterResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:PingResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="ping">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="pong" nillable="true"
type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="pingResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:PingResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="logout">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                    </xs:sequence>
                </xs:complexType>
```

```
            </xs:element>
            <xs:element name="logoutResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:RepositoryResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="login">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="credentials" nillable="true"
type="ax214:RepositoryCredentials"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="loginResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:RepositoryLoginResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="loadDocument">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="docs" nillable="true"
type="ax214:DocumentList"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="loadDocumentResponse">
                <xs:complexType>
                    <xs:sequence>
```

```
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:LoadDocumentResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="invokeMethod">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="methodName" nillable="true"
type="xs:string"/>
                        <xs:element minOccurs="0" name="args" nillable="true"
type="ax214:RepositoryPropArray"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="invokeMethodResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:DynamicResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getViewAsOptions">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getViewAsOptionsResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:ViewAsOptionsResult"/>
                    </xs:sequence>
```

```
                </xs:complexType>
            </xs:element>
            <xs:element name="getTreeView">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="tree" nillable="true"
type="ax214:TreeRequest"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getTreeViewResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:TreeResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getSupportedOptions">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getSupportedOptionsResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:OptionsResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getPushNotifications">
                <xs:complexType>
```

```
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getPushNotificationsResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:NotificationsResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getCustomTree">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getCustomTreeResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:TreeResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getAboutInfo">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
```

```
            <xs:element name="getAboutInfoResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:AboutResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="findApplications">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="filter" nillable="true"
type="ax214:ApplicationFilter"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="findApplicationsResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:ApplicationListResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="advancedSearch">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="session" nillable="true"
type="ax214:RepositorySession"/>
                        <xs:element minOccurs="0" name="criteria" nillable="true"
type="ax214:AdvancedSearchCriteria"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="advancedSearchResponse">
                <xs:complexType>
```

```
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax214:AdvancedSearchResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:schema>
        <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://api.v2.repository.xenos.com/xsd">
            <xs:complexType name="RepositoryRequest">
                <xs:sequence>
                    <xs:element minOccurs="0" name="properties" nillable="true"
type="ax213:RepositoryPropMap"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="RepositoryPropMap">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="names"
nillable="true" type="xs:string"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="values"
nillable="true" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="RepositorySession">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="instanceName"
nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="timeout" type="xs:long"/>
                            <xs:element minOccurs="0" name="token" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="user" nillable="true"
type="xs:string"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="TreeRequest">
```

```
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="tree" nillable="true"
type="ax213:TreeView"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="TreeView">
                <xs:sequence>
                    <xs:element minOccurs="0" name="expression" nillable="true"
type="xs:string"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="nodes"
nillable="true" type="ax213:RepositoryTreeNode"/>
                    <xs:element minOccurs="0" name="path" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="RepositoryTreeNode">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="children"
nillable="true" type="ax213:RepositoryTreeNode"/>
                    <xs:element minOccurs="0" name="hasToc" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="id" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="isApplication" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="leaf" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="name" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="RepositoryResult">
                <xs:sequence>
                    <xs:element minOccurs="0" name="message" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="properties" nillable="true"
type="ax213:RepositoryPropMap"/>
                    <xs:element minOccurs="0" name="serverTime" type="xs:long"/>
                    <xs:element minOccurs="0" name="status" type="xs:string"/>
```

```
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="InstancesResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="instances" nillable="true" type="ax213:RepositoryInstance"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="RepositoryInstance">
                <xs:sequence>
                    <xs:element minOccurs="0" name="description" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="name" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="options" nillable="true"
type="ax213:RepositoryOptions"/>
                    <xs:element minOccurs="0" name="type" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="RepositoryOptions">
                <xs:sequence>
                    <xs:element minOccurs="0" name="canFilterApplications"
type="xs:boolean"/>
                    <xs:element minOccurs="0" name="canLoad" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="canRetrieveMultipleDocuments"
type="xs:boolean"/>
                    <xs:element minOccurs="0" name="canSearch" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="canTimeoutClient" type="xs:boolean"/
>
                    <xs:element minOccurs="0" name="hasAdvancedSearch"
type="xs:boolean"/>
                    <xs:element minOccurs="0" name="hasCustomTree" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="hasPushNotifications"
type="xs:boolean"/>
                    <xs:element minOccurs="0" name="hasVersioning" type="xs:boolean"/>
```

```
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="SearchCriteria">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="andSearch"
type="xs:boolean"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="applicationIds" nillable="true" type="xs:string"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="criteria" nillable="true" type="ax213:SearchItem"/>
                            <xs:element minOccurs="0" name="interval" type="xs:int"/>
                            <xs:element minOccurs="0" name="limit" type="xs:int"/>
                            <xs:element minOccurs="0" name="options" nillable="true"
type="ax213:RepositoryPropMap"/>
                            <xs:element minOccurs="0" name="sortOrder" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="start" type="xs:int"/>
                            <xs:element minOccurs="0" name="vars" nillable="true"
type="ax213:RepositoryPropMap"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="SearchItem">
                <xs:sequence>
                    <xs:element minOccurs="0" name="intervalAttribute"
type="xs:boolean"/>
                    <xs:element minOccurs="0" name="limitAttribute" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="name" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="operator" nillable="true"
type="ax213:Operator"/>
                    <xs:element minOccurs="0" name="sortAttribute" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="specialAttribute" type="xs:boolean"/
>
                    <xs:element minOccurs="0" name="startAttribute" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="tokenAttribute" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="value1" nillable="true"
type="xs:string"/>
```

```
                    <xs:element minOccurs="0" name="value2" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="Operator">
                <xs:sequence>
                    <xs:element minOccurs="0" name="id" type="xs:int"/>
                    <xs:element minOccurs="0" name="name" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="shortName" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="HitList">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="allHits" nillable="true" type="ax213:HitItem"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="attribNames" nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="document" nillable="true"
type="ax213:DocumentResult"/>
                            <xs:element minOccurs="0" name="folderName" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="totalHits" type="xs:int"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="HitItem">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="attribValues"
nillable="true" type="xs:string"/>
                    <xs:element minOccurs="0" name="docId" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="docName" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="documentLength" type="xs:long"/>
                    <xs:element minOccurs="0" name="extension" nillable="true"
type="xs:string"/>
```

```
                    <xs:element minOccurs="0" name="folderName" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="implContext" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="numberOfSections" type="xs:int"/>
                    <xs:element minOccurs="0" name="resourceId" nillable="true"
type="xs:string"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0"
name="retrieveTypes" nillable="true" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="DocumentMetaResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="applicationName"
nillable="true" type="xs:string"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="documentIds" nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="ext" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="metaInfo" nillable="true"
type="ax213:RepositoryPropMap"/>
                            <xs:element minOccurs="0" name="resourceData"
nillable="true" type="xs:base64Binary"/>
                            <xs:element minOccurs="0" name="resourceId" nillable="true"
type="xs:string"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="selectedDocIds" nillable="true" type="xs:int"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="DocumentResult">
                <xs:complexContent>
                    <xs:extension base="ax213:DocumentMetaResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="docBytes" nillable="true"
type="xs:base64Binary"/>
                        </xs:sequence>
                    </xs:extension>
```

```
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="FieldQueryRequest">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" minOccurs="0" name="ids"
nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="retrieveDocProperties"
type="xs:boolean"/>
                            <xs:element minOccurs="0" name="retrieveFileProperties"
type="xs:boolean"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="FieldQueryResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="fields" nillable="true" type="ax213:RepositoryField"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="RepositoryField">
                <xs:sequence>
                    <xs:element minOccurs="0" name="fieldName" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="fieldType" type="xs:string"/>
                    <xs:element minOccurs="0" name="format" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="symbolicName" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="Document">
                <xs:complexContent>
```

```
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="applicationId"
nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="docId" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="resourceId" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="retrieveAllSections"
type="xs:boolean"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="RetrieveOptions">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="convertToExt"
nillable="true" type="xs:string"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="TocRequest">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="direction" type="xs:string"/
>
                            <xs:element minOccurs="0" name="length" type="xs:int"/>
                            <xs:element minOccurs="0" name="orderBy" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="path" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="start" type="xs:int"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
```

```
            </xs:complexType>
            <xs:complexType name="TocResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="defaultExt" nillable="true"
type="xs:string"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="fields" nillable="true" type="ax213:RepositoryField"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="tocItems" nillable="true" type="ax213:TocItem"/>
                            <xs:element minOccurs="0" name="totalRows" type="xs:int"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="TocItem">
                <xs:sequence>
                    <xs:element minOccurs="0" name="applicationId" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="docId" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="docName" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="ext" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="numberOfSections" type="xs:int"/>
                    <xs:element minOccurs="0" name="row" type="xs:int"/>
                    <xs:element minOccurs="0" name="size" type="xs:long"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="values"
nillable="true" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="PingResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="result" nillable="true"
type="xs:string"/>
                        </xs:sequence>
```

```
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="RepositoryCredentials">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="instanceName"
nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="login" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="password" nillable="true"
type="xs:string"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="RepositoryLoginResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="changedPassword"
type="xs:boolean"/>
                            <xs:element minOccurs="0" name="session" nillable="true"
type="ax213:RepositorySession"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="DocumentList">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence/>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="LoadDocumentResult">
                <xs:complexContent>
```

```
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence/>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="RepositoryPropArray">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="values"
nillable="true" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="DynamicResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="results" nillable="true"
type="ax213:RepositoryPropMap"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="ViewAsOptionsResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="inExts" nillable="true"
type="ax213:RepositoryPropArray"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="outExts" nillable="true" type="ax213:RepositoryPropArray"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="TreeResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
```

```
                            <xs:element minOccurs="0" name="tree" nillable="true"
type="ax213:TreeView"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="OptionsResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="options" nillable="true"
type="ax213:RepositoryOptions"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="NotificationsResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="instance" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="notificationTime"
type="xs:long"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="notifications" nillable="true" type="ax213:Notification"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="Notification">
                <xs:sequence>
                    <xs:element minOccurs="0" name="disposition" type="xs:int"/>
                    <xs:element minOccurs="0" name="id" type="xs:int"/>
                    <xs:element minOccurs="0" name="message" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="messageType" type="xs:int"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="parts"
nillable="true" type="xs:string"/>
```

```
                    <xs:element minOccurs="0" name="timestamp" type="xs:long"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="AboutResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="aboutMap" nillable="true"
type="ax213:RepositoryPropMap"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="ApplicationFilter">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="direction" type="xs:string"/
>
                            <xs:element minOccurs="0" name="expression" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="isCustomView"
type="xs:boolean"/>
                            <xs:element minOccurs="0" name="length" type="xs:int"/>
                            <xs:element minOccurs="0" name="start" type="xs:int"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="ApplicationListResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="applications" nillable="true" type="ax213:ApplicationItem"/>
                            <xs:element minOccurs="0" name="expression" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="totalRows" type="xs:int"/>
```

```
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="ApplicationItem">
                <xs:sequence>
                    <xs:element minOccurs="0" name="hasToc" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="id" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="name" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="path" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="AdvancedSearchCriteria">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryRequest">
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="applications" nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="datePattern"
nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="dateRecent" type="xs:int"/>
                            <xs:element minOccurs="0" name="dateType" type="xs:int"/>
                            <xs:element minOccurs="0" name="endDate" type="xs:long"/>
                            <xs:element minOccurs="0" name="ext" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="length" type="xs:int"/>
                            <xs:element minOccurs="0" name="orderBy" nillable="true"
type="xs:string"/>
                            <xs:element minOccurs="0" name="orderDir" type="xs:string"/>
                            <xs:element minOccurs="0" name="searchCriteria"
nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="searchType" type="xs:int"/>
                            <xs:element minOccurs="0" name="start" type="xs:int"/>
                            <xs:element minOccurs="0" name="startDate" type="xs:long"/>
                        </xs:sequence>
                    </xs:extension>
```

```
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="AdvancedSearchResult">
                <xs:complexContent>
                    <xs:extension base="ax213:RepositoryResult">
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="allHits" nillable="true" type="ax213:AdvancedHitItem"/>
                            <xs:element maxOccurs="unbounded" minOccurs="0"
name="attribNames" nillable="true" type="xs:string"/>
                            <xs:element minOccurs="0" name="searchType" type="xs:int"/>
                            <xs:element minOccurs="0" name="totalRows" type="xs:int"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
            <xs:complexType name="AdvancedHitItem">
                <xs:sequence>
                    <xs:element minOccurs="0" name="applicationName" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="applicationPath" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="docId" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="docName" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="fileId" type="xs:int"/>
                    <xs:element minOccurs="0" name="fileName" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="ingestionDate" type="xs:long"/>
                    <xs:element minOccurs="0" name="numberOfSections" type="xs:int"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0"
name="propertyNames" nillable="true" type="xs:string"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0"
name="propertyValues" nillable="true" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="logoutRequest">
```

```
        <wsdl:part name="parameters" element="ns:logout"/>
    </wsdl:message>
    <wsdl:message name="logoutResponse">
        <wsdl:part name="parameters" element="ns:logoutResponse"/>
    </wsdl:message>
    <wsdl:message name="RepositoryWebServiceException">
        <wsdl:part name="parameters" element="ns:RepositoryWebServiceException"/>
    </wsdl:message>
    <wsdl:message name="getViewAsOptionsRequest">
        <wsdl:part name="parameters" element="ns:getViewAsOptions"/>
    </wsdl:message>
    <wsdl:message name="getViewAsOptionsResponse">
        <wsdl:part name="parameters" element="ns:getViewAsOptionsResponse"/>
    </wsdl:message>
    <wsdl:message name="loadDocumentRequest">
        <wsdl:part name="parameters" element="ns:loadDocument"/>
    </wsdl:message>
    <wsdl:message name="loadDocumentResponse">
        <wsdl:part name="parameters" element="ns:loadDocumentResponse"/>
    </wsdl:message>
    <wsdl:message name="retrieveFieldsRequest">
        <wsdl:part name="parameters" element="ns:retrieveFields"/>
    </wsdl:message>
    <wsdl:message name="retrieveFieldsResponse">
        <wsdl:part name="parameters" element="ns:retrieveFieldsResponse"/>
    </wsdl:message>
    <wsdl:message name="retrieveDocumentMetaDataRequest">
        <wsdl:part name="parameters" element="ns:retrieveDocumentMetaData"/>
    </wsdl:message>
    <wsdl:message name="retrieveDocumentMetaDataResponse">
        <wsdl:part name="parameters" element="ns:retrieveDocumentMetaDataResponse"/>
    </wsdl:message>
    <wsdl:message name="retrieveDocumentWithOptionsRequest">
        <wsdl:part name="parameters" element="ns:retrieveDocumentWithOptions"/>
    </wsdl:message>
```

```
    <wsdl:message name="retrieveDocumentWithOptionsResponse">
        <wsdl:part name="parameters" element="ns:retrieveDocumentWithOptionsResponse"/>
    </wsdl:message>
    <wsdl:message name="getPushNotificationsRequest">
        <wsdl:part name="parameters" element="ns:getPushNotifications"/>
    </wsdl:message>
    <wsdl:message name="getPushNotificationsResponse">
        <wsdl:part name="parameters" element="ns:getPushNotificationsResponse"/>
    </wsdl:message>
    <wsdl:message name="getSupportedOptionsRequest">
        <wsdl:part name="parameters" element="ns:getSupportedOptions"/>
    </wsdl:message>
    <wsdl:message name="getSupportedOptionsResponse">
        <wsdl:part name="parameters" element="ns:getSupportedOptionsResponse"/>
    </wsdl:message>
    <wsdl:message name="advancedSearchRequest">
        <wsdl:part name="parameters" element="ns:advancedSearch"/>
    </wsdl:message>
    <wsdl:message name="advancedSearchResponse">
        <wsdl:part name="parameters" element="ns:advancedSearchResponse"/>
    </wsdl:message>
    <wsdl:message name="retrieveDocumentRequest">
        <wsdl:part name="parameters" element="ns:retrieveDocument"/>
    </wsdl:message>
    <wsdl:message name="retrieveDocumentResponse">
        <wsdl:part name="parameters" element="ns:retrieveDocumentResponse"/>
    </wsdl:message>
    <wsdl:message name="findApplicationsRequest">
        <wsdl:part name="parameters" element="ns:findApplications"/>
    </wsdl:message>
    <wsdl:message name="findApplicationsResponse">
        <wsdl:part name="parameters" element="ns:findApplicationsResponse"/>
    </wsdl:message>
    <wsdl:message name="setCustomTreeRequest">
        <wsdl:part name="parameters" element="ns:setCustomTree"/>
```

```
    </wsdl:message>
    <wsdl:message name="setCustomTreeResponse">
        <wsdl:part name="parameters" element="ns:setCustomTreeResponse"/>
    </wsdl:message>
    <wsdl:message name="pingAdapterRequest">
        <wsdl:part name="parameters" element="ns:pingAdapter"/>
    </wsdl:message>
    <wsdl:message name="pingAdapterResponse">
        <wsdl:part name="parameters" element="ns:pingAdapterResponse"/>
    </wsdl:message>
    <wsdl:message name="getTreeViewRequest">
        <wsdl:part name="parameters" element="ns:getTreeView"/>
    </wsdl:message>
    <wsdl:message name="getTreeViewResponse">
        <wsdl:part name="parameters" element="ns:getTreeViewResponse"/>
    </wsdl:message>
    <wsdl:message name="pingRequest">
        <wsdl:part name="parameters" element="ns:ping"/>
    </wsdl:message>
    <wsdl:message name="pingResponse">
        <wsdl:part name="parameters" element="ns:pingResponse"/>
    </wsdl:message>
    <wsdl:message name="loginRequest">
        <wsdl:part name="parameters" element="ns:login"/>
    </wsdl:message>
    <wsdl:message name="loginResponse">
        <wsdl:part name="parameters" element="ns:loginResponse"/>
    </wsdl:message>
    <wsdl:message name="getAboutInfoRequest">
        <wsdl:part name="parameters" element="ns:getAboutInfo"/>
    </wsdl:message>
    <wsdl:message name="getAboutInfoResponse">
        <wsdl:part name="parameters" element="ns:getAboutInfoResponse"/>
    </wsdl:message>
    <wsdl:message name="getCustomTreeRequest">
```

```
        <wsdl:part name="parameters" element="ns:getCustomTree"/>
    </wsdl:message>
    <wsdl:message name="getCustomTreeResponse">
        <wsdl:part name="parameters" element="ns:getCustomTreeResponse"/>
    </wsdl:message>
    <wsdl:message name="invokeMethodRequest">
        <wsdl:part name="parameters" element="ns:invokeMethod"/>
    </wsdl:message>
    <wsdl:message name="invokeMethodResponse">
        <wsdl:part name="parameters" element="ns:invokeMethodResponse"/>
    </wsdl:message>
    <wsdl:message name="retrieveInstancesRequest">
        <wsdl:part name="parameters" element="ns:retrieveInstances"/>
    </wsdl:message>
    <wsdl:message name="retrieveInstancesResponse">
        <wsdl:part name="parameters" element="ns:retrieveInstancesResponse"/>
    </wsdl:message>
    <wsdl:message name="retrieveHitListRequest">
        <wsdl:part name="parameters" element="ns:retrieveHitList"/>
    </wsdl:message>
    <wsdl:message name="retrieveHitListResponse">
        <wsdl:part name="parameters" element="ns:retrieveHitListResponse"/>
    </wsdl:message>
    <wsdl:message name="readTocRequest">
        <wsdl:part name="parameters" element="ns:readToc"/>
    </wsdl:message>
    <wsdl:message name="readTocResponse">
        <wsdl:part name="parameters" element="ns:readTocResponse"/>
    </wsdl:message>
    <wsdl:portType name="RepositoryWebServicePortType">
        <wsdl:operation name="logout">
            <wsdl:input message="ns:logoutRequest" wsaw:Action="urn:logout"/>
            <wsdl:output message="ns:logoutResponse" wsaw:Action="urn:logoutResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:logoutRepositoryWebServiceException"/>
```

```
        </wsdl:operation>
        <wsdl:operation name="getViewAsOptions">
            <wsdl:input message="ns:getViewAsOptionsRequest"
wsaw:Action="urn:getViewAsOptions"/>
            <wsdl:output message="ns:getViewAsOptionsResponse"
wsaw:Action="urn:getViewAsOptionsResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:getViewAsOptionsRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="loadDocument">
            <wsdl:input message="ns:loadDocumentRequest" wsaw:Action="urn:loadDocument"/
>
            <wsdl:output message="ns:loadDocumentResponse"
wsaw:Action="urn:loadDocumentResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:loadDocumentRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="retrieveFields">
            <wsdl:input message="ns:retrieveFieldsRequest"
wsaw:Action="urn:retrieveFields"/>
            <wsdl:output message="ns:retrieveFieldsResponse"
wsaw:Action="urn:retrieveFieldsResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:retrieveFieldsRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocumentMetaData">
            <wsdl:input message="ns:retrieveDocumentMetaDataRequest"
wsaw:Action="urn:retrieveDocumentMetaData"/>
            <wsdl:output message="ns:retrieveDocumentMetaDataResponse"
wsaw:Action="urn:retrieveDocumentMetaDataResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:retrieveDocumentMetaDataRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocumentWithOptions">
            <wsdl:input message="ns:retrieveDocumentWithOptionsRequest"
wsaw:Action="urn:retrieveDocumentWithOptions"/>
            <wsdl:output message="ns:retrieveDocumentWithOptionsResponse"
wsaw:Action="urn:retrieveDocumentWithOptionsResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:retrieveDocumentWithOptionsRepositoryWebServiceException"/>
```

```
        </wsdl:operation>
        <wsdl:operation name="getPushNotifications">
            <wsdl:input message="ns:getPushNotificationsRequest"
wsaw:Action="urn:getPushNotifications"/>
            <wsdl:output message="ns:getPushNotificationsResponse"
wsaw:Action="urn:getPushNotificationsResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:getPushNotificationsRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="getSupportedOptions">
            <wsdl:input message="ns:getSupportedOptionsRequest"
wsaw:Action="urn:getSupportedOptions"/>
            <wsdl:output message="ns:getSupportedOptionsResponse"
wsaw:Action="urn:getSupportedOptionsResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:getSupportedOptionsRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="advancedSearch">
            <wsdl:input message="ns:advancedSearchRequest"
wsaw:Action="urn:advancedSearch"/>
            <wsdl:output message="ns:advancedSearchResponse"
wsaw:Action="urn:advancedSearchResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:advancedSearchRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocument">
            <wsdl:input message="ns:retrieveDocumentRequest"
wsaw:Action="urn:retrieveDocument"/>
            <wsdl:output message="ns:retrieveDocumentResponse"
wsaw:Action="urn:retrieveDocumentResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:retrieveDocumentRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="findApplications">
            <wsdl:input message="ns:findApplicationsRequest"
wsaw:Action="urn:findApplications"/>
            <wsdl:output message="ns:findApplicationsResponse"
wsaw:Action="urn:findApplicationsResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:findApplicationsRepositoryWebServiceException"/>
```

```
        </wsdl:operation>
        <wsdl:operation name="setCustomTree">
            <wsdl:input message="ns:setCustomTreeRequest"
wsaw:Action="urn:setCustomTree"/>
            <wsdl:output message="ns:setCustomTreeResponse"
wsaw:Action="urn:setCustomTreeResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:setCustomTreeRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="pingAdapter">
            <wsdl:input message="ns:pingAdapterRequest" wsaw:Action="urn:pingAdapter"/>
            <wsdl:output message="ns:pingAdapterResponse"
wsaw:Action="urn:pingAdapterResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:pingAdapterRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="getTreeView">
            <wsdl:input message="ns:getTreeViewRequest" wsaw:Action="urn:getTreeView"/>
            <wsdl:output message="ns:getTreeViewResponse"
wsaw:Action="urn:getTreeViewResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:getTreeViewRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="ping">
            <wsdl:input message="ns:pingRequest" wsaw:Action="urn:ping"/>
            <wsdl:output message="ns:pingResponse" wsaw:Action="urn:pingResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException" wsaw:Action="urn:pingRepositoryWebServiceException"/
>
        </wsdl:operation>
        <wsdl:operation name="login">
            <wsdl:input message="ns:loginRequest" wsaw:Action="urn:login"/>
            <wsdl:output message="ns:loginResponse" wsaw:Action="urn:loginResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:loginRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="getAboutInfo">
            <wsdl:input message="ns:getAboutInfoRequest" wsaw:Action="urn:getAboutInfo"/
>
```

```
            <wsdl:output message="ns:getAboutInfoResponse"
wsaw:Action="urn:getAboutInfoResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:getAboutInfoRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="getCustomTree">
            <wsdl:input message="ns:getCustomTreeRequest"
wsaw:Action="urn:getCustomTree"/>
            <wsdl:output message="ns:getCustomTreeResponse"
wsaw:Action="urn:getCustomTreeResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:getCustomTreeRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="invokeMethod">
            <wsdl:input message="ns:invokeMethodRequest" wsaw:Action="urn:invokeMethod"/
>
            <wsdl:output message="ns:invokeMethodResponse"
wsaw:Action="urn:invokeMethodResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:invokeMethodRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="retrieveInstances">
            <wsdl:input message="ns:retrieveInstancesRequest"
wsaw:Action="urn:retrieveInstances"/>
            <wsdl:output message="ns:retrieveInstancesResponse"
wsaw:Action="urn:retrieveInstancesResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:retrieveInstancesRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="retrieveHitList">
            <wsdl:input message="ns:retrieveHitListRequest"
wsaw:Action="urn:retrieveHitList"/>
            <wsdl:output message="ns:retrieveHitListResponse"
wsaw:Action="urn:retrieveHitListResponse"/>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:retrieveHitListRepositoryWebServiceException"/>
        </wsdl:operation>
        <wsdl:operation name="readToc">
            <wsdl:input message="ns:readTocRequest" wsaw:Action="urn:readToc"/>
```

```
            <wsdl:output message="ns:readTocResponse" wsaw:Action="urn:readTocResponse"/
>
            <wsdl:fault message="ns:RepositoryWebServiceException"
name="RepositoryWebServiceException"
wsaw:Action="urn:readTocRepositoryWebServiceException"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="RepositoryWebServiceSoap11Binding"
type="ns:RepositoryWebServicePortType">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/
>
        <wsdl:operation name="logout">
            <soap:operation soapAction="urn:logout" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="loadDocument">
            <soap:operation soapAction="urn:loadDocument" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="getViewAsOptions">
            <soap:operation soapAction="urn:getViewAsOptions" style="document"/>
            <wsdl:input>
```

```
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="retrieveFields">
            <soap:operation soapAction="urn:retrieveFields" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocumentMetaData">
            <soap:operation soapAction="urn:retrieveDocumentMetaData" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocumentWithOptions">
```

```
            <soap:operation soapAction="urn:retrieveDocumentWithOptions"
style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="getSupportedOptions">
            <soap:operation soapAction="urn:getSupportedOptions" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="getPushNotifications">
            <soap:operation soapAction="urn:getPushNotifications" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
```

```
        </wsdl:operation>
        <wsdl:operation name="advancedSearch">
            <soap:operation soapAction="urn:advancedSearch" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocument">
            <soap:operation soapAction="urn:retrieveDocument" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="findApplications">
            <soap:operation soapAction="urn:findApplications" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
```

```
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="setCustomTree">
            <soap:operation soapAction="urn:setCustomTree" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="pingAdapter">
            <soap:operation soapAction="urn:pingAdapter" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="ping">
            <soap:operation soapAction="urn:ping" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
```

```
            <soap:fault use="literal" name="RepositoryWebServiceException"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getTreeView">
        <soap:operation soapAction="urn:getTreeView" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="RepositoryWebServiceException">
            <soap:fault use="literal" name="RepositoryWebServiceException"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="login">
        <soap:operation soapAction="urn:login" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="RepositoryWebServiceException">
            <soap:fault use="literal" name="RepositoryWebServiceException"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="getAboutInfo">
        <soap:operation soapAction="urn:getAboutInfo" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
```

```
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="getCustomTree">
            <soap:operation soapAction="urn:getCustomTree" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="invokeMethod">
            <soap:operation soapAction="urn:invokeMethod" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="retrieveInstances">
            <soap:operation soapAction="urn:retrieveInstances" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
```

```
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="retrieveHitList">
            <soap:operation soapAction="urn:retrieveHitList" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="readToc">
            <soap:operation soapAction="urn:readToc" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="RepositoryWebServiceException">
                <soap:fault use="literal" name="RepositoryWebServiceException"/>
            </wsdl:fault>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="RepositoryWebServiceHttpBinding"
type="ns:RepositoryWebServicePortType">
        <http:binding verb="POST"/>
        <wsdl:operation name="logout">
            <http:operation location="logout"/>
```

```
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="loadDocument">
            <http:operation location="loadDocument"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getViewAsOptions">
            <http:operation location="getViewAsOptions"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="retrieveFields">
            <http:operation location="retrieveFields"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocumentMetaData">
```

```
            <http:operation location="retrieveDocumentMetaData"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocumentWithOptions">
            <http:operation location="retrieveDocumentWithOptions"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getSupportedOptions">
            <http:operation location="getSupportedOptions"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getPushNotifications">
            <http:operation location="getPushNotifications"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
```

```
        <wsdl:operation name="advancedSearch">
            <http:operation location="advancedSearch"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="retrieveDocument">
            <http:operation location="retrieveDocument"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="findApplications">
            <http:operation location="findApplications"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="setCustomTree">
            <http:operation location="setCustomTree"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
```

```
        </wsdl:operation>
        <wsdl:operation name="pingAdapter">
            <http:operation location="pingAdapter"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="ping">
            <http:operation location="ping"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getTreeView">
            <http:operation location="getTreeView"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="login">
            <http:operation location="login"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
```

```
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getAboutInfo">
            <http:operation location="getAboutInfo"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getCustomTree">
            <http:operation location="getCustomTree"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="invokeMethod">
            <http:operation location="invokeMethod"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="retrieveInstances">
            <http:operation location="retrieveInstances"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
```

```
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="retrieveHitList">
            <http:operation location="retrieveHitList"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="readToc">
            <http:operation location="readToc"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="RepositoryWebService">
        <wsdl:port name="RepositoryWebServiceHttpSoap11Endpoint"
binding="ns:RepositoryWebServiceSoap11Binding">
            <soap:address location="http://localhost:8080/axis2/services/
RepositoryWebService.RepositoryWebServiceHttpSoap11Endpoint/"/>
        </wsdl:port>
        <wsdl:port name="RepositoryWebServiceHttpEndpoint"
binding="ns:RepositoryWebServiceHttpBinding">
            <http:address location="http://localhost:8080/axis2/services/
RepositoryWebService.RepositoryWebServiceHttpEndpoint/"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

5.2.2 Repository Services 2 Sample Request

5.2.2.1 Sample 1

The following is a sample SOAP request generated while trying to retrieve a HitList:

```
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ws="http://ws.v2.repository.xenos.com" xmlns:xsd="http://
api.v2.repository.xenos.com/xsd">
    <soapenv:Header/>
    <soapenv:Body>
       <ws:retrieveHitList>
          <ws:session>
             <xsd:instanceName>esre3Test</xsd:instanceName>
             <xsd:token>esre.user=ablack=770368921-535274218-1474990662</xsd:token>
          </ws:session>
          <ws:criteria>
             <xsd:andSearch>true</xsd:andSearch>
             <xsd:applicationIds>BestBankStatements</xsd:applicationIds>
             <xsd:criteria>
                <xsd:name>FullName</xsd:name>
                <xsd:operator>
                   <xsd:id>7</xsd:id>
                   <xsd:name>OP_LIKE</xsd:name>
                   <xsd:shortName>LI</xsd:shortName>
                </xsd:operator>
                <xsd:value1>%</xsd:value1>
             </xsd:criteria>
             <xsd:interval>25</xsd:interval>
             <xsd:limit>100</xsd:limit>
            <xsd:start>0</xsd:start>
         </ws:criteria>
       </ws:retrieveHitList>
    </soapenv:Body>
 </soapenv:Envelope>
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
       <ns:retrieveHitListResponse xmlns:ns="http://ws.v2.repository.xenos.com">
          <ns:return xsi:type="ax213:HitList" xmlns:ax213="http://
api.v2.repository.xenos.com/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
             <ax213:message xsi:nil="true"/>
             <ax213:properties xsi:nil="true"/>
             <ax213:serverTime>101</ax213:serverTime>
             <ax213:status>s</ax213:status>
             <ax213:allHits xsi:type="ax213:HitItem">
                <ax213:attribValues>20120723000000</ax213:attribValues>
                <ax213:attribValues>4555-9999-0001</ax213:attribValues>
                <ax213:attribValues>200909</ax213:attribValues>
                <ax213:attribValues>April Black</ax213:attribValues>
                <ax213:docId>"1:0~$REPORTID$=BestBankStatements~"</ax213:docId>
                <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
                <ax213:documentLength>0</ax213:documentLength>
                <ax213:extension>pdf</ax213:extension>
                <ax213:folderName>BestBankStatements</ax213:folderName>
                <ax213:implContext xsi:nil="true"/>
                <ax213:numberOfSections>1</ax213:numberOfSections>
                <ax213:resourceId xsi:nil="true"/>
                <ax213:retrieveTypes xsi:nil="true"/>
             </ax213:allHits>
             <ax213:allHits xsi:type="ax213:HitItem">
                <ax213:attribValues>20120723000000</ax213:attribValues>
                <ax213:attribValues>4555-9999-0002</ax213:attribValues>
                <ax213:attribValues>200909</ax213:attribValues>
                <ax213:attribValues>Marie Cortez</ax213:attribValues>
                <ax213:docId>"1:1~$REPORTID$=BestBankStatements~"</ax213:docId>
                <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
                <ax213:documentLength>0</ax213:documentLength>
                <ax213:extension>pdf</ax213:extension>
                <ax213:folderName>BestBankStatements</ax213:folderName>
```

```
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0003</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Deepinder Mitwar</ax213:attribValues>
            <ax213:docId>"1:2~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0004</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Peter Mcmillan</ax213:attribValues>
            <ax213:docId>"1:3~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0005</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Mudassar Nazar</ax213:attribValues>
            <ax213:docId>"1:4~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0006</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Belle Harmon</ax213:attribValues>
            <ax213:docId>"1:5~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0007</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Andreas Figueroa</ax213:attribValues>
            <ax213:docId>"1:6~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
```

```
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0008</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Mariam Kemp</ax213:attribValues>
            <ax213:docId>"1:7~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0009</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Candace Wallace</ax213:attribValues>
            <ax213:docId>"1:8~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0011</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Gabina Montano</ax213:attribValues>
            <ax213:docId>"1:9~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0012</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Evangeline Morrow</ax213:attribValues>
            <ax213:docId>"1:10~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0013</ax213:attribValues>
```

```
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Stewart Valdez</ax213:attribValues>
            <ax213:docId>"1:11~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0014</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Seth Soto</ax213:attribValues>
            <ax213:docId>"1:12~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0015</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Maxine Powers</ax213:attribValues>
            <ax213:docId>"1:13~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120723000000</ax213:attribValues>
            <ax213:attribValues>4555-9999-0017</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>Cameron Fisher</ax213:attribValues>
            <ax213:docId>"1:14~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20110809_1411_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120730153939</ax213:attribValues>
            <ax213:attribValues>4555-9999-0001</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>zApril Black</ax213:attribValues>
            <ax213:docId>"8:0~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
```

```
            </ax213:allHits>
            <ax213:allHits xsi:type="ax213:HitItem">
               <ax213:attribValues>20120730153939</ax213:attribValues>
               <ax213:attribValues>4555-9999-0002</ax213:attribValues>
               <ax213:attribValues>200909</ax213:attribValues>
               <ax213:attribValues>zMarie Cortez</ax213:attribValues>
               <ax213:docId>"8:1~$REPORTID$=BestBankStatements~"</ax213:docId>
               <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
               <ax213:documentLength>0</ax213:documentLength>
               <ax213:extension>pdf</ax213:extension>
               <ax213:folderName>BestBankStatements</ax213:folderName>
               <ax213:implContext xsi:nil="true"/>
               <ax213:numberOfSections>1</ax213:numberOfSections>
               <ax213:resourceId xsi:nil="true"/>
               <ax213:retrieveTypes xsi:nil="true"/>
            </ax213:allHits>
            <ax213:allHits xsi:type="ax213:HitItem">
               <ax213:attribValues>20120730153939</ax213:attribValues>
               <ax213:attribValues>4555-9999-0003</ax213:attribValues>
               <ax213:attribValues>200909</ax213:attribValues>
               <ax213:attribValues>zDeepinder Mitwar</ax213:attribValues>
               <ax213:docId>"8:2~$REPORTID$=BestBankStatements~"</ax213:docId>
               <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
               <ax213:documentLength>0</ax213:documentLength>
               <ax213:extension>pdf</ax213:extension>
               <ax213:folderName>BestBankStatements</ax213:folderName>
               <ax213:implContext xsi:nil="true"/>
               <ax213:numberOfSections>1</ax213:numberOfSections>
               <ax213:resourceId xsi:nil="true"/>
               <ax213:retrieveTypes xsi:nil="true"/>
            </ax213:allHits>
            <ax213:allHits xsi:type="ax213:HitItem">
               <ax213:attribValues>20120730153939</ax213:attribValues>
               <ax213:attribValues>4555-9999-0004</ax213:attribValues>
              <ax213:attribValues>200909</ax213:attribValues>
               <ax213:attribValues>zPeter Mcmillan</ax213:attribValues>
               <ax213:docId>"8:3~$REPORTID$=BestBankStatements~"</ax213:docId>
               <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
               <ax213:documentLength>0</ax213:documentLength>
               <ax213:extension>pdf</ax213:extension>
               <ax213:folderName>BestBankStatements</ax213:folderName>
               <ax213:implContext xsi:nil="true"/>
               <ax213:numberOfSections>1</ax213:numberOfSections>
               <ax213:resourceId xsi:nil="true"/>
               <ax213:retrieveTypes xsi:nil="true"/>
            </ax213:allHits>
            <ax213:allHits xsi:type="ax213:HitItem">
               <ax213:attribValues>20120730153939</ax213:attribValues>
               <ax213:attribValues>4555-9999-0005</ax213:attribValues>
               <ax213:attribValues>200909</ax213:attribValues>
               <ax213:attribValues>zMudassar Nazar</ax213:attribValues>
               <ax213:docId>"8:4~$REPORTID$=BestBankStatements~"</ax213:docId>
               <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
               <ax213:documentLength>0</ax213:documentLength>
               <ax213:extension>pdf</ax213:extension>
               <ax213:folderName>BestBankStatements</ax213:folderName>
               <ax213:implContext xsi:nil="true"/>
               <ax213:numberOfSections>1</ax213:numberOfSections>
               <ax213:resourceId xsi:nil="true"/>
               <ax213:retrieveTypes xsi:nil="true"/>
            </ax213:allHits>
            <ax213:allHits xsi:type="ax213:HitItem">
               <ax213:attribValues>20120730153939</ax213:attribValues>
               <ax213:attribValues>4555-9999-0006</ax213:attribValues>
               <ax213:attribValues>200909</ax213:attribValues>
               <ax213:attribValues>zBelle Harmon</ax213:attribValues>
               <ax213:docId>"8:5~$REPORTID$=BestBankStatements~"</ax213:docId>
               <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
               <ax213:documentLength>0</ax213:documentLength>
               <ax213:extension>pdf</ax213:extension>
               <ax213:folderName>BestBankStatements</ax213:folderName>
```

```
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120730153939</ax213:attribValues>
            <ax213:attribValues>4555-9999-0007</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>zAndreas Figueroa</ax213:attribValues>
            <ax213:docId>"8:6~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120730153939</ax213:attribValues>
            <ax213:attribValues>4555-9999-0008</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>zMariam Kemp</ax213:attribValues>
            <ax213:docId>"8:7~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120730153939</ax213:attribValues>
            <ax213:attribValues>4555-9999-0009</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>zCandace Wallace</ax213:attribValues>
            <ax213:docId>"8:8~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:allHits xsi:type="ax213:HitItem">
            <ax213:attribValues>20120730153939</ax213:attribValues>
            <ax213:attribValues>4555-9999-0011</ax213:attribValues>
            <ax213:attribValues>200909</ax213:attribValues>
            <ax213:attribValues>zGabina Montano</ax213:attribValues>
            <ax213:docId>"8:9~$REPORTID$=BestBankStatements~"</ax213:docId>
            <ax213:docName>Stmt_20120730_1539_001.pdf</ax213:docName>
            <ax213:documentLength>0</ax213:documentLength>
            <ax213:extension>pdf</ax213:extension>
            <ax213:folderName>BestBankStatements</ax213:folderName>
            <ax213:implContext xsi:nil="true"/>
            <ax213:numberOfSections>1</ax213:numberOfSections>
            <ax213:resourceId xsi:nil="true"/>
           <ax213:retrieveTypes xsi:nil="true"/>
         </ax213:allHits>
         <ax213:attribNames>$LOADDATE$</ax213:attribNames>
         <ax213:attribNames>AccountNumber</ax213:attribNames>
         <ax213:attribNames>StatementDate</ax213:attribNames>
         <ax213:attribNames>FullName</ax213:attribNames>
         <ax213:document xsi:nil="true"/>
         <ax213:folderName>BestBankStatements</ax213:folderName>
```

```
            <ax213:totalHits>75</ax213:totalHits>
         </ns:return>
      </ns:retrieveHitListResponse>
   </soapenv:Body>
 </soapenv:Envelope>
```

5.2.2.2 Sample 2

The following is a sample SOAP request generated while trying to retrieve a document:

```
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ws="http://ws.v2.repository.xenos.com" xmlns:xsd="http://
api.v2.repository.xenos.com/xsd">
   <soapenv:Header/>
   <soapenv:Body>
      <ws:retrieveDocumentWithOptions>
         <ws:session>
            <xsd:instanceName>esre3Test</xsd:instanceName>
            <xsd:token>esre.user=ablack=770368921-535274218-1474990662</xsd:token>
         </ws:session>
         <ws:docs>
            <xsd:applicationId>BestBankStatements</xsd:applicationId>
            <xsd:docId>"1:1~$REPORTID$=BestBankStatements~"</xsd:docId>
         </ws:docs>
         <ws:options>
            <!--Optional:-->
            <xsd:convertToExt></xsd:convertToExt>
         </ws:options>
      </ws:retrieveDocumentWithOptions>
   </soapenv:Body>
 </soapenv:Envelope>
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <ns:retrieveDocumentWithOptionsResponse xmlns:ns="http://
ws.v2.repository.xenos.com">
         <ns:return xsi:type="ax213:DocumentResult" xmlns:ax213="http://
api.v2.repository.xenos.com/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <ax213:message xsi:nil="true"/>
            <ax213:properties xsi:nil="true"/>
            <ax213:serverTime>633</ax213:serverTime>
            <ax213:status>s</ax213:status>
```

```
            <ax213:applicationName>BestBankStatements</ax213:applicationName>
            <ax213:documentIds>"1:1:2~"</ax213:documentIds>
            <ax213:ext>pdf</ax213:ext>
            <ax213:metaInfo xsi:type="ax213:RepositoryPropMap">
               <ax213:names>$REPORTID$</ax213:names>
               <ax213:values>BestBankStatements</ax213:values>
            </ax213:metaInfo>
            <ax213:resourceData xsi:nil="true"/>
            <ax213:resourceId xsi:nil="true"/>
            <ax213:selectedDocIds>0</ax213:selectedDocIds>

           <ax213:docBytes>JVBERi0xLjQKJeLjz9MKMyAwIG9iago8PAovRmlsdGVyIFsvRmxhdGVEZWNvZ
GVdIAovTGVuZ3RoIDM1ODIKPj4gc3RyZWFtCnicjVptcxu3Ef7uX8EZclpmatN4B64z/
UBJdMxaIlWRju1MZjKORMtsRFKlJDvpry8Whzvs4kClUZIRtM89WCweLLC44z3mf3j4//W295/4K
+9JN6p6VrERdxwsrz/y3tm+96/eC4Sxqic1H1kZISJCWABc/dh7wdVI6973njQ97ZG8t+1xxuLvd71FQqQ/
b3uisiNbAmHD1pPyMsz3phyLvYn4e9abaCFAWpVAonItSBrh+8IgN
+J1X64nmRhpD9JaenPdqkEB4FuGVyMVaKDB9EgmgFMjOwLqBgVUJgBgRrax4QEuAbhSHt9QxBbl4Np4/
xuS2KIsgin/fMMSW5RFcOOfaFhiK2MxhMWUWKz1YWpZ6hZlkULjqNUtyiJl8iT8njE4jWISWxlDUFPLUbcoi/
+QP5fo7i1OGCVa/A312+KU9yZ1OdZH2gz9ZfZH9TqD2TDu+O/rWwf845eW55OIODSm5gFQuoX4G4qKk3TYD7SZR/
Ef9LH9S/yVDuHzP6OVH/zjb5XYBX8v6aCrlWZNTv
```

… truncated …

```
 </ns:return>
      </ns:retrieveDocumentWithOptionsResponse>
   </soapenv:Body>
 </soapenv:Envelope>
```

5.2.3 Repository Services 2 Sample Login Request and Response

The following is a sample SOAP login request and response:

```
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ws="http://ws.v2.repository.xenos.com" xmlns:xsd="http://
api.v2.repository.xenos.com/xsd">
   <soapenv:Header/>
   <soapenv:Body>
      <ws:login>
         <ws:credentials>
            <xsd:instanceName>esre3Test</xsd:instanceName>
            <xsd:login>ablack</xsd:login>
            <xsd:password>Xenos@123</xsd:password>
         </ws:credentials>
      </ws:login>
```

```
    </soapenv:Body>
 </soapenv:Envelope>
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
       <ns:loginResponse xmlns:ns="http://ws.v2.repository.xenos.com">
          <ns:return xsi:type="ax213:RepositoryLoginResult" xmlns:ax213="http://
api.v2.repository.xenos.com/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
             <ax213:message xsi:nil="true"/>
             <ax213:properties xsi:nil="true"/>
             <ax213:serverTime>0</ax213:serverTime>
             <ax213:status>s</ax213:status>
             <ax213:changedPassword>false</ax213:changedPassword>
             <ax213:session xsi:type="ax213:RepositorySession">
                <ax213:properties xsi:nil="true"/>
                <ax213:instanceName>esre3Test</ax213:instanceName>
                <ax213:timeout>0</ax213:timeout>
                <ax213:token>esre.user=ablack=770368921-535274218-1474990662</
ax213:token>
                <ax213:user>ablack</ax213:user>
             </ax213:session>
          </ns:return>
       </ns:loginResponse>
    </soapenv:Body>
 </soapenv:Envelope>
```

5.3 Axis2 JobRunner

This section demonstrates job submission using requests from the Axis2 JobRunner.

5.3.1 Axis2 JobRunner WSDL

The WSDL document for the Axis2 JobRunner web service interface is located here:

`http://localhost:8080/axis2/services/JobRunnerService?wsdl`

The contents of the WSDL document are as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns="http://jobrunner.service.ws.framework.xenos.com"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ax232="http://serialization.ws2.j2ee.xenos.com/xsd"
xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:ax231="http://serialization.j2ee.xenos.com/xsd"
```

```
xmlns:ax236="http://rmi.java/xsd"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:ax237="http://io.java/xsd" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://jobrunner.service.ws.framework.xenos.com">
    <wsdl:documentation>JobRunnerService</wsdl:documentation>
    <wsdl:types>
        <xs:schema xmlns:ax238="http://io.java/xsd" attributeFormDefault="qualified"
elementFormDefault="qualified" targetNamespace="http://rmi.java/xsd">
            <xs:import namespace="http://io.java/xsd"/>
            <xs:complexType name="RemoteException">
                <xs:complexContent>
                    <xs:extension base="ax238:IOException">
                        <xs:sequence>
                            <xs:element minOccurs="0" name="cause" nillable="true"
type="xs:anyType"/>
                            <xs:element minOccurs="0" name="message" nillable="true"
type="xs:string"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
        </xs:schema>
        <xs:schema xmlns:ax233="http://serialization.ws2.j2ee.xenos.com/xsd"
attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://
serialization.j2ee.xenos.com/xsd">
            <xs:import namespace="http://serialization.ws2.j2ee.xenos.com/xsd"/>
            <xs:complexType name="RequestOptions">
                <xs:sequence>
                    <xs:element minOccurs="0" name="filter" type="xs:boolean"/>
                    <xs:element minOccurs="0" name="inputIncludes" nillable="true"
type="ax232:JobVarMap"/>
                    <xs:element minOccurs="0" name="resultIncludes" nillable="true"
type="ax232:JobVarMap"/>
                    <xs:element minOccurs="0" name="token" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="variableIncludes" nillable="true"
type="ax232:JobVarMap"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="MapData">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="keys"
nillable="true" type="xs:string"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="values"
nillable="true" type="xs:anyType"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="LoginResultWrapper">
                <xs:sequence>
                    <xs:element minOccurs="0" name="message" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="properties" nillable="true"
type="ax231:MapData"/>
                    <xs:element minOccurs="0" name="status" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="tokenID" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:schema>
        <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://serialization.ws2.j2ee.xenos.com/xsd">
            <xs:complexType name="JobVarMap">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="keys"
nillable="true" type="xs:string"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="values"
nillable="true" type="ax232:JobVarMapEntry"/>
```

```
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="JobVarMapEntry">
                <xs:sequence>
                    <xs:element minOccurs="0" name="value" nillable="true"
type="xs:string"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="values"
nillable="true" type="ax232:JobVarMap"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="XDocMap">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="XDocData"
nillable="true" type="ax232:XDocWrapper"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="XDocKeys"
nillable="true" type="xs:string"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="multiXDocData"
nillable="true" type="ax232:MultiXDocWrapper"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="multiXDocKeys"
nillable="true" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="XDocWrapper">
                <xs:sequence>
                    <xs:element minOccurs="0" name="data" nillable="true"
type="xs:base64Binary"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="MultiXDocWrapper">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="multiXDocData"
nillable="true" type="ax232:XDocWrapper"/>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="multiXDocKeys"
nillable="true" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="JobRunnerResult">
                <xs:sequence>
                    <xs:element minOccurs="0" name="alias" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="component" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="errorNumber" type="xs:long"/>
                    <xs:element minOccurs="0" name="failureMessage" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="fatalErrorNumber" type="xs:long"/>
                    <xs:element minOccurs="0" name="inputMap" nillable="true"
type="ax232:XDocMap"/>
                    <xs:element minOccurs="0" name="inputSize" type="xs:long"/>
                    <xs:element minOccurs="0" name="invocationMethod" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="jobId" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="jobVarMap" nillable="true"
type="ax232:JobVarMap"/>
                    <xs:element minOccurs="0" name="outputSize" type="xs:long"/>
                    <xs:element minOccurs="0" name="queuedMs" type="xs:long"/>
                    <xs:element minOccurs="0" name="resultMap" nillable="true"
type="ax232:XDocMap"/>
                    <xs:element minOccurs="0" name="runningMs" type="xs:long"/>
                    <xs:element minOccurs="0" name="startTime" nillable="true"
type="xs:dateTime"/>
                    <xs:element minOccurs="0" name="status" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="stopTime" nillable="true"
type="xs:dateTime"/>
                    <xs:element minOccurs="0" name="suspendMs" type="xs:long"/>
                    <xs:element minOccurs="0" name="totalMs" type="xs:long"/>
                    <xs:element minOccurs="0" name="userField" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="warningNumber" type="xs:long"/>
```

```
                </xs:sequence>
            </xs:complexType>
        </xs:schema>
                <xs:schema attributeFormDefault="qualified"
elementFormDefault="qualified" targetNamespace="http://io.java/xsd">
            <xs:complexType name="IOException">
                <xs:sequence/>
            </xs:complexType>
        </xs:schema>
        <xs:schema xmlns:ax239="http://rmi.java/xsd" xmlns:ax234="http://
serialization.j2ee.xenos.com/xsd" xmlns:ax235="http://serialization.ws2.j2ee.xenos.com/
xsd" attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://jobrunner.service.ws.framework.xenos.com">
            <xs:import namespace="http://serialization.j2ee.xenos.com/xsd"/>
            <xs:import namespace="http://serialization.ws2.j2ee.xenos.com/xsd"/>
            <xs:import namespace="http://rmi.java/xsd"/>
            <xs:element name="runJob">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="request" nillable="true"
type="ax231:RequestOptions"/>
                        <xs:element minOccurs="0" name="alias" nillable="true"
type="xs:string"/>
                        <xs:element minOccurs="0" name="jobVars" nillable="true"
type="ax232:JobVarMap"/>
                        <xs:element minOccurs="0" name="inputs" nillable="true"
type="ax232:XDocMap"/>
                        <xs:element minOccurs="0" name="outputs" nillable="true"
type="ax232:XDocMap"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="runJobResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax232:JobRunnerResult"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="JobRunnerServiceRemoteException">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="RemoteException" nillable="true"
type="ax236:RemoteException"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="logout">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="token" nillable="true"
type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="logoutResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" type="xs:boolean"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="authenticate">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="user" nillable="true"
type="xs:string"/>
                        <xs:element minOccurs="0" name="password" nillable="true"
type="xs:string"/>
                        <xs:element minOccurs="0" name="loginProperties" nillable="true"
```

```
type="ax231:MapData"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="authenticateResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
type="ax231:LoginResultWrapper"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="authenticateRequest">
        <wsdl:part name="parameters" element="ns:authenticate"/>
    </wsdl:message>
    <wsdl:message name="authenticateResponse">
        <wsdl:part name="parameters" element="ns:authenticateResponse"/>
    </wsdl:message>
    <wsdl:message name="JobRunnerServiceRemoteException">
        <wsdl:part name="parameters" element="ns:JobRunnerServiceRemoteException"/>
    </wsdl:message>
    <wsdl:message name="logoutRequest">
        <wsdl:part name="parameters" element="ns:logout"/>
    </wsdl:message>
    <wsdl:message name="logoutResponse">
        <wsdl:part name="parameters" element="ns:logoutResponse"/>
    </wsdl:message>
    <wsdl:message name="runJobRequest">
        <wsdl:part name="parameters" element="ns:runJob"/>
    </wsdl:message>
    <wsdl:message name="runJobResponse">
        <wsdl:part name="parameters" element="ns:runJobResponse"/>
    </wsdl:message>
    <wsdl:portType name="JobRunnerServicePortType">
        <wsdl:operation name="authenticate">
            <wsdl:input message="ns:authenticateRequest" wsaw:Action="urn:authenticate"/>
            <wsdl:output message="ns:authenticateResponse"
wsaw:Action="urn:authenticateResponse"/>
            <wsdl:fault message="ns:JobRunnerServiceRemoteException"
name="JobRunnerServiceRemoteException"
wsaw:Action="urn:authenticateJobRunnerServiceRemoteException"/>
        </wsdl:operation>
        <wsdl:operation name="logout">
            <wsdl:input message="ns:logoutRequest" wsaw:Action="urn:logout"/>
            <wsdl:output message="ns:logoutResponse" wsaw:Action="urn:logoutResponse"/>
            <wsdl:fault message="ns:JobRunnerServiceRemoteException"
name="JobRunnerServiceRemoteException"
wsaw:Action="urn:logoutJobRunnerServiceRemoteException"/>
        </wsdl:operation>
        <wsdl:operation name="runJob">
            <wsdl:input message="ns:runJobRequest" wsaw:Action="urn:runJob"/>
            <wsdl:output message="ns:runJobResponse" wsaw:Action="urn:runJobResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="JobRunnerServiceSoap11Binding"
type="ns:JobRunnerServicePortType">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
        <wsdl:operation name="logout">
            <soap:operation soapAction="urn:logout" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="JobRunnerServiceRemoteException">
                <soap:fault use="literal" name="JobRunnerServiceRemoteException"/>
            </wsdl:fault>
        </wsdl:operation>
```

```
        <wsdl:operation name="authenticate">
            <soap:operation soapAction="urn:authenticate" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="JobRunnerServiceRemoteException">
                <soap:fault use="literal" name="JobRunnerServiceRemoteException"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="runJob">
            <soap:operation soapAction="urn:runJob" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="JobRunnerServiceHttpBinding"
type="ns:JobRunnerServicePortType">
        <http:binding verb="POST"/>
        <wsdl:operation name="logout">
            <http:operation location="logout"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="authenticate">
            <http:operation location="authenticate"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="runJob">
            <http:operation location="runJob"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="JobRunnerService">
        <wsdl:port name="JobRunnerServiceHttpSoap11Endpoint"
binding="ns:JobRunnerServiceSoap11Binding">
            <soap:address location="http://localhost:8080/axis2/services/
JobRunnerService.JobRunnerServiceHttpSoap11Endpoint/"/>
        </wsdl:port>
        <wsdl:port name="JobRunnerServiceHttpEndpoint"
binding="ns:JobRunnerServiceHttpBinding">
            <http:address location="http://localhost:8080/axis2/services/
JobRunnerService.JobRunnerServiceHttpEndpoint/"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

5.3.2 Axis2 JobRunner Sample Request

The following is a sample SOAP request generated while trying to execute a sample process flow:

Note: To reduce the size of the response, you can filter the request with specific values, which is illustrated in "Axis2 JobRunner Sample Filtered Request" on page 148.

```
<!-- This is an example of using the JobRunner to execute the sample pdf2tiff process
already included with Process Manager
     Results are not filtered so runJob by default will return everything in the job
ticket -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:job="http://jobrunner.service.ws.framework.xenos.com" xmlns:xsd="http://
serialization.j2ee.xenos.com/xsd" xmlns:xsd1="http://serialization.ws2.j2ee.xenos.com/
xsd">
   <soapenv:Header/>
   <soapenv:Body>
      <job:runJob>

         <job:request>
            <!--REQUIRED: specify token returned by authenticate operation.  User
                must be someone with JobRunner privileges (i.e. JobRunner) -->

<xsd:token>7e47e373beb9d06123714e59f40ddaa67d37f303699be8a3839b807a32ab449b</xsd:token>

         </job:request>
         <!-- The component to run relative to base repository (i.e. Process Manager
process flow or Document Transform project) -->
         <!-- Here we are using the sample pdf2tiff process flow already provided with
Process Manager install-->

<job:alias>_sample/OutputTransformationServer/OutputTransformation/processFlow/Sample-
pdf2tiff.xProcessFlow</job:alias>

         <!--Optional: Specify job variables to set here-->
         <job:jobVars>
            <!--Job variable names:-->
            <xsd1:keys>MyJobVar1</xsd1:keys>
            <xsd1:keys>MyJobVar2</xsd1:keys>
            <xsd1:keys>MyJobVar3</xsd1:keys>

            <!--Job variable values, values assigned in same order as above :-->
            <xsd1:values>
               <xsd1:value>One</xsd1:value>
               <xsd1:values/>
            </xsd1:values>

            <xsd1:values>
               <xsd1:value>Two</xsd1:value>
               <xsd1:values/>
            </xsd1:values>

            <xsd1:values>
               <xsd1:value>Three</xsd1:value>
               <xsd1:values/>
            </xsd1:values>

         </job:jobVars>

         <!--Input to the job -->
         <job:inputs>
            <!--Actual data used as input -->
            <xsd1:XDocData>
               <!--Here we are providing a Base64 encoded version of <baseRepository>/
```

```
_sample/OutputTransformation/applications/sample/input/sample.pdf-->
               <xsd1:data>
                 <!-- base 64 encoded data goes here -->
               </xsd1:data>
            </xsd1:XDocData>
            <!--Specify the inputMap key to store this data.  In this example, Sample-
pdf2tiff.xProcessFlow expects data to come from the DataToConvert InputMap key -->
            <xsd1:XDocKeys>DataToConvert</xsd1:XDocKeys>

         </job:inputs>


      </job:runJob>
   </soapenv:Body>
</soapenv:Envelope>
```

5.3.3 Axis2 JobRunner Sample Response

The following is a sample SOAP response generated after a sample process flow was executed:

Note: This sample has been trimmed to save space, but represents all the main fields that the server should return. Be aware that by default, the Base64-encoded values of both the input and output data are included as part of the response. To reduce the size of the response, you can filter the response with specific values, which is illustrated in "Axis2 JobRunner Sample Filtered Response" on page 150.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <ns:runJobResponse xmlns:ns="http://jobrunner.service.ws.framework.xenos.com">
         <ns:return xsi:type="ax232:JobRunnerResult" xmlns:ax232="http://
serialization.ws2.j2ee.xenos.com/xsd" xmlns:ax231="http://serialization.j2ee.xenos.com/
xsd" xmlns:ax236="http://rmi.java/xsd" xmlns:ax237="http://io.java/xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <ax232:alias/>

<ax232:component>_sample/OutputTransformationServer/OutputTransformation/processFlow/
Sample-pdf2tiff.xProcessFlow</ax232:component>
            <ax232:errorNumber>0</ax232:errorNumber>
            <ax232:failureMessage/>
            <ax232:fatalErrorNumber>0</ax232:fatalErrorNumber>
            <ax232:inputMap xsi:type="ax232:XDocMap">
               <ax232:XDocData xsi:type="ax232:XDocWrapper">
                  <ax232:data>
                  <!-- base 64 encoded input appears here, removed to improve
readability -->
                  </ax232:data>
               </ax232:XDocData>
               <ax232:XDocKeys>DataToConvert</ax232:XDocKeys>
               <ax232:multiXDocData xsi:nil="true"/>
               <ax232:multiXDocKeys xsi:nil="true"/>
            </ax232:inputMap>
            <ax232:inputSize>130936</ax232:inputSize>
            <ax232:invocationMethod/>
            <ax232:jobId>20150710_1447_001</ax232:jobId>
            <ax232:jobVarMap xsi:type="ax232:JobVarMap">
               <ax232:keys>java.vendor</ax232:keys>
               <ax232:keys>d2eInputFilePageCount</ax232:keys>
               <ax232:keys>sun.java.launcher</ax232:keys>
               <ax232:keys>catalina.base</ax232:keys>
               <ax232:keys>sun.management.compiler</ax232:keys>
               <ax232:keys>xenos.base.repository</ax232:keys>
               <ax232:keys>catalina.useNaming</ax232:keys>
```

```
                <ax232:keys>os.name</ax232:keys>
                <ax232:keys>sun.boot.class.path</ax232:keys>
                <ax232:keys>java.util.logging.config.file</ax232:keys>
                <ax232:keys>sun.desktop</ax232:keys>
                <ax232:keys>java.vm.specification.vendor</ax232:keys>
                <ax232:keys>java.runtime.version</ax232:keys>
                <ax232:keys>base.repository.dir</ax232:keys>

                <ax232:keys>d2eOutputPath</ax232:keys>
                <ax232:keys>d2eJobId</ax232:keys>

<!— additional keys from response removed to improve readability -->

                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>Sun Microsystems Inc.</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>3O</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>SUN_STANDARD</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>C:\apache-tomcat-7.0.56</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>HotSpot 64-Bit Tiered Compilers</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>/OpenText/OTS/baseRepository</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>true</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>Windows 7</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>C:\Program Files\Java\jdk1.6.0_31\jre\lib\resources.jar;C:
\Program Files\Java\jdk1.6.0_31\jre\lib\rt.jar;C:\Program Files\Java\jdk1.6.0_31\jre\lib
\sunrsasign.jar;C:\Program Files\Java\jdk1.6.0_31\jre\lib\jsse.jar;C:\Program Files\Java
\jdk1.6.0_31\jre\lib\jce.jar;C:\Program Files\Java\jdk1.6.0_31\jre\lib\charsets.jar;C:
\Program Files\Java\jdk1.6.0_31\jre\lib\modules\jdk.boot.jar;C:\Program Files\Java
\jdk1.6.0_31\jre\classes</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>C:\apache-tomcat-7.0.56\conf\logging.properties</
ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>windows</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>Sun Microsystems Inc.</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
                <ax232:values xsi:type="ax232:JobVarMapEntry">
                   <ax232:value>1.6.0_31-b05</ax232:value>
                   <ax232:values xsi:nil="true"/>
                </ax232:values>
```

```
               <ax232:values xsi:type="ax232:JobVarMapEntry">
                  <ax232:value>/OTS/baseRepository</ax232:value>
                  <ax232:values xsi:nil="true"/>
               </ax232:values>

<!— additional keys from response removed to improve readability -->

            </ax232:jobVarMap>
            <ax232:outputSize>939070</ax232:outputSize>
            <ax232:queuedMs>17</ax232:queuedMs>
            <ax232:resultMap xsi:type="ax232:XDocMap">
               <ax232:XDocData xsi:type="ax232:XDocWrapper">
                  <ax232:data xsi:nil="true"/>
               </ax232:XDocData>
               <ax232:XDocKeys>XDoc</ax232:XDocKeys>
               <ax232:multiXDocData xsi:type="ax232:MultiXDocWrapper">
                  <ax232:multiXDocData xsi:type="ax232:XDocWrapper">
                     <ax232:data>
                     <!-- base 64 encoded result data will appear here, removed to
improve readability -->
                     </ax232:data>
                  </ax232:multiXDocData>

<ax232:multiXDocKeys>C:\OTS\baseRepository\common\_configs\/../output/sample-
pdf2tif-1.tif</ax232:multiXDocKeys>
               </ax232:multiXDocData>
               <ax232:multiXDocKeys>constantPrint.ComponentList.TIFF
Generator.Parms.FdOutput</ax232:multiXDocKeys>
            </ax232:resultMap>
            <ax232:runningMs>3834</ax232:runningMs>
            <ax232:startTime>2015-07-10T14:47:50.499-04:00</ax232:startTime>
            <ax232:status>FINISHED</ax232:status>
            <ax232:stopTime>2015-07-10T14:47:54.350-04:00</ax232:stopTime>
            <ax232:suspendMs>0</ax232:suspendMs>
            <ax232:totalMs>3851</ax232:totalMs>
            <ax232:userField/>
            <ax232:warningNumber>1</ax232:warningNumber>
         </ns:return>
      </ns:runJobResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

5.3.4 Axis2 JobRunner Sample Filtered Request

The following is a sample SOAP request generated while executing a sample process flow. This is the same example as the previous request, but with filtering applied, which causes the response to return only the desired ResultMap names and job variables you request.

Note: To see a sample request without filtering enabled, see “Axis2 JobRunner Sample Request” on page 145.

```
<!-- This is an example of using the JobRunner to execute the sample pdf2tiff process
already included with Process Manager
     The results are filtered by setting the filter boolean parameter to true and
specifying additional filter parameters -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:job="http://jobrunner.service.ws.framework.xenos.com" xmlns:xsd="http://
serialization.j2ee.xenos.com/xsd" xmlns:xsd1="http://serialization.ws2.j2ee.xenos.com/
xsd">
   <soapenv:Header/>
   <soapenv:Body>
      <job:runJob>

         <job:request>
```

```
        <!-- Default value is false.  If set to true, runJob will filter the results
sent back to you based
             on the filter parameters below -->
         <xsd:filter>true</xsd:filter>

               <!--REQUIRED: specify token returned by authenticate operation.  User
                must be someone with JobRunner privileges (i.e. JobRunner)  -->

<xsd:token>7e47e373beb9d06123714e59f40ddaa67d37f303699be8a3839b807a32ab449b</xsd:token>

            <!-- Specify which ResultMap keys you want returned.  In this example, we
grab just the ResultMap key for the Tiff Generator -->
            <xsd:resultIncludes>
               <xsd1:keys>constantPrint.ComponentList.TIFF Generator.Parms.FdOutput</
xsd1:keys>
            </xsd:resultIncludes>

            <!-- Specify the Job variables you want returned as part of the runJob
response.-->
            <xsd:variableIncludes>
             <xsd1:keys>MyJobVar1</xsd1:keys>
             <xsd1:keys>MyJobVar2</xsd1:keys>
             <xsd1:keys>MyJobVar3</xsd1:keys>
            </xsd:variableIncludes>

         </job:request>
         <!-- The component to run relative to base repository (i.e. Process Manager
process flow or Document Transform project) -->
         <!-- Here we are using the sample pdf2tiff process flow already provided with
Process Manager install-->

<job:alias>_sample/OutputTransformationServer/OutputTransformation/processFlow/Sample-
pdf2tiff.xProcessFlow</job:alias>

         <!--Optional: Specify job variables to set here-->
         <job:jobVars>
            <!--Job variable names:-->
            <xsd1:keys>MyJobVar1</xsd1:keys>
            <xsd1:keys>MyJobVar2</xsd1:keys>
            <xsd1:keys>MyJobVar3</xsd1:keys>

            <!--Job variable values, values assigned in same order as above :-->
            <xsd1:values>
               <xsd1:value>One</xsd1:value>
               <xsd1:values/>
            </xsd1:values>

            <xsd1:values>
               <xsd1:value>Two</xsd1:value>
               <xsd1:values/>
            </xsd1:values>

            <xsd1:values>
               <xsd1:value>Three</xsd1:value>
               <xsd1:values/>
            </xsd1:values>

         </job:jobVars>

         <!--Input to the job -->
         <job:inputs>
            <!--Actual data used as input -->
            <xsd1:XDocData>
               <!--Here we are providing a Base64 encoded version of <baseRepository>/
_sample/d2eVision/applications/sample/input/sample.pdf-->
               <xsd1:data>

                 <!-- base 64 encoded data goes here -->

</xsd1:data>
```

```
            </xsd1:XDocData>
            <!--Specify the inputMap key to store this data.  In this example, Sample-
pdf2tiff.xProcessFlow expects data to come from the DataToConvert InputMap key -->
            <xsd1:XDocKeys>DataToConvert</xsd1:XDocKeys>

         </job:inputs>
      </job:runJob>
   </soapenv:Body>
</soapenv:Envelope>
```

5.3.5 Axis2 JobRunner Sample Filtered Response

The following is a sample SOAP response after submitting the filtered request shown in "Axis2 JobRunner Sample Response" on page 146. Due to the filtering parameters used, this response only contains the job variables (MyJobVar1, MyJobVar2, and MyJobVar3) and the data from the result mapping key (constantPrint.ComponentList.TIFFGenerator.Parms.FdOutput).

Note: To see a sample response without filtering enabled, see "Axis2 JobRunner Sample Response" on page 146.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <ns:runJobResponse xmlns:ns="http://jobrunner.service.ws.framework.xenos.com">
         <ns:return xsi:type="ax232:JobRunnerResult" xmlns:ax232="http://
serialization.ws2.j2ee.xenos.com/xsd" xmlns:ax231="http://serialization.j2ee.xenos.com/
xsd" xmlns:ax236="http://rmi.java/xsd" xmlns:ax237="http://io.java/xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

            <ax232:alias/>

<ax232:component>_sample/OutputTransformationServer/OutputTransformation/processFlow/
Sample-pdf2tiff.xProcessFlow</ax232:component>
            <ax232:errorNumber>0</ax232:errorNumber>
            <ax232:failureMessage/>
            <ax232:fatalErrorNumber>0</ax232:fatalErrorNumber>
            <ax232:inputMap xsi:type="ax232:XDocMap">
               <ax232:XDocData xsi:nil="true"/>
               <ax232:XDocKeys xsi:nil="true"/>
               <ax232:multiXDocData xsi:nil="true"/>
               <ax232:multiXDocKeys xsi:nil="true"/>
            </ax232:inputMap>
            <ax232:inputSize>130936</ax232:inputSize>
            <ax232:invocationMethod/>
            <ax232:jobId>20150710_1515_002</ax232:jobId>
            <ax232:jobVarMap xsi:type="ax232:JobVarMap">
               <ax232:keys>MyJobVar3</ax232:keys>
               <ax232:keys>MyJobVar2</ax232:keys>
               <ax232:keys>MyJobVar1</ax232:keys>
               <ax232:values xsi:type="ax232:JobVarMapEntry">
                  <ax232:value>Three</ax232:value>
                  <ax232:values xsi:nil="true"/>
               </ax232:values>
               <ax232:values xsi:type="ax232:JobVarMapEntry">
                  <ax232:value>Two</ax232:value>
                  <ax232:values xsi:nil="true"/>
               </ax232:values>
               <ax232:values xsi:type="ax232:JobVarMapEntry">
                  <ax232:value>One</ax232:value>
                  <ax232:values xsi:nil="true"/>
               </ax232:values>
            </ax232:jobVarMap>
            <ax232:outputSize>939070</ax232:outputSize>
            <ax232:queuedMs>1</ax232:queuedMs>
            <ax232:resultMap xsi:type="ax232:XDocMap">
```

```
               <ax232:XDocData xsi:nil="true"/>
               <ax232:XDocKeys xsi:nil="true"/>
               <ax232:multiXDocData xsi:type="ax232:MultiXDocWrapper">
                  <ax232:multiXDocData xsi:type="ax232:XDocWrapper">
                     <ax232:data>
  <!-- base 64 encoded data appears here, removed for readability -->
                     </ax232:data>
                  </ax232:multiXDocData>

<ax232:multiXDocKeys>C:\OTS\baseRepository\common\_configs\/../output/sample-
pdf2tif-1.tif</ax232:multiXDocKeys>
               </ax232:multiXDocData>
               <ax232:multiXDocKeys>constantPrint.ComponentList.TIFF
Generator.Parms.FdOutput</ax232:multiXDocKeys>
            </ax232:resultMap>
            <ax232:runningMs>1443</ax232:runningMs>
            <ax232:startTime>2015-07-10T15:16:16.059-04:00</ax232:startTime>
            <ax232:status>FINISHED</ax232:status>
            <ax232:stopTime>2015-07-10T15:16:17.503-04:00</ax232:stopTime>
            <ax232:suspendMs>0</ax232:suspendMs>
            <ax232:totalMs>1444</ax232:totalMs>
            <ax232:userField/>
            <ax232:warningNumber>1</ax232:warningNumber>
         </ns:return>
      </ns:runJobResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Chapter 6

Working with JavaScript and the JavaScriptProcess Component

The JavaScriptProcess component provides users with a means of including scripts for interacting with a running process. API users can employ the component to add custom logic to their processes using JavaScript instead of writing, compiling, and managing custom Java-based components.

Scripts can be written at the parameter or component level, but share some common principles, functionality, and environment variables.

JavaScript and EcmaScript

The terms JavaScript and EcmaScript are often used interchangeably. By definition, EcmaScript refers to the scripting language based on the international ECMA-262 specification.

JavaScript is an implementation of EcmaScript so online documentation, tutorials, and code samples exist using both names and either can be referenced. For more information on compatible versions of these scripting languages and some additional resources, see "JavaScript Compatibility" on page 159.

6.1 Configuring the JavaScriptProcess Component

The Script component is available under the **Processes** folder in the **Palette** and can be inserted into your process flows. It can be added as an inline, single use instance within the process flow or created as an external standalone component (with the `.xJavaScript` process file extension), which can be shared between multiple process flows.

Parameter Scripting

Parameter scripting allows the use of JavaScript to dynamically resolve the values of any configuration parameters. This is sometimes required when configuration options are dependent on some external criteria and cannot be set statically.

In addition to the standard wizards, menus, and embedded editors, most configuration parameters can also be accessed through the standard **Properties** window. Here, you can open a Property dialog box by clicking a parameter's respective ellipsis button. Then in the Property dialog box, you can activate the embedded **JavaScript Editor** from the drop-down menu.

Using the JavaScript Editor, you can edit your source code while the editor offers syntax highlighting and some basic code completion.

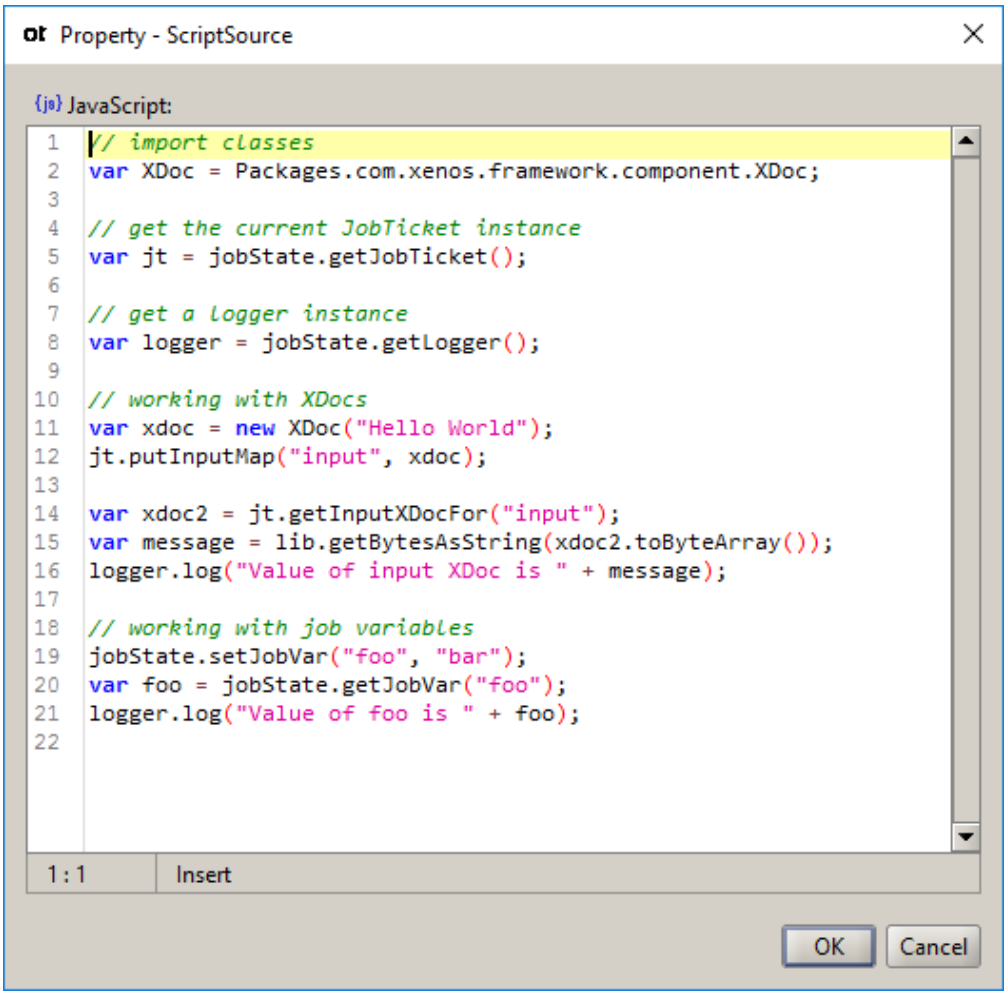

Figure 6-1: JavaScript Editor instance with JavaScriptProcess component sample

Component Scripting

The JavaScriptProcess component provides users with a means of including scripts for interacting with a running process. API users can employ the component to add custom logic to their processes using JavaScript instead of writing, compiling, and managing custom Java-based components.

6.2 Standard Bindings

When JavaScript functions are executed, a set of variables called bindings are made available to the script as variables. These binding variables provide a mechanism for passing data and objects to and from the script. The bindings are Java objects that can be seamlessly accessed in the JavaScript code the same way you would interact with any JavaScript variable or object.

The standard set of bindings are described below. You can also learn more about these Java binding objects by referring to the corresponding Javadoc documentation in the OpenText Output Transformation Suite help system.

6.2.1 jobState

The jobState object is the primary mechanism for interacting with the running process and underlying environment. It provides access to job variables and other stateful objects.

The underlying jobState Java object available to you in your script is dependent on the context on which the script is executed. This allows scripts to access state objects which are relevant for their application. For example, jobState will provide access to the current page object (XifPage) and any text fields that have been located.

There is a base Java class that defines methods available to all types of job states, as well as various subclasses that provide additional functionality. These subclasses inherit all methods and properties of the base class. Some code snippets showing sample usage are also shown below.

Content	Corresponding Java Class Name	Details
Default	`com.xenos.framework.util.jobstate.JobState`	The base job state. Provides access to properties such as job variables, job ID, active logger, and I/O utilities.
Output Transformation Designer	`com.xenos.framework.component.RunnableJobState`	In addition to the properties provided by the default job state, this Java class offers access to the JobTicket object for the running process.

```
//Access and use the active logger.  Expect output to be logged to the corresponding log for your job
var logger = jobState.getLogger();
logger.log("Here is a message for the log!");

try {
        //The job ID for the associated job.  Could be a Process Designer job, or Doc. Transform job
        var jobId = jobState.getJobId();

        //Every instance of JobState has a unique, numeric ID.  Useful for debugging, etc.
        var jobStateId = jobState.getJobStateId();

        //Access job variables which are strings.  Optionally provide a default value
        var defaultFileName = "defaultTestIndex.txt";
        var indexFileName = jobState.getJobVar("indexFileName"); // if not defined, returns empty string ""
        var indexFileName = jobState.getJobVar("indexFileName", defaultFileName);

        //Access job variables which are objects
        var customerObject = jobState.getJobVarObject("Customer");
        //Depending on what type of object it is, you can call the available methods.
        var customerName = customerObject.getName();

        //Access the IoUtility instance
        var ioUtility = jobState.getIoUtility();
        var inputStream = ioUtility.createInputStream("{NOCRLF}", indexFileName, "");

        //Replace job variables.  If variables are not defined, the place holders remain in the string
        var stringWithJobVars = "Here is a string that contains a job variables ${myJobVar} and ${myJobVar2}";
        var stringWithoutJobVars = jobState.replaceJobVariables(stringWithJobVariables);
}
catch (exception) {
        // Catch any possible exception
        logger.log("Error in script: " + exception);
}
```

Figure 6-2: Sample usage of default JobState

```
// Get a logger instance
var logger = jobState.getLogger();

try {
        // Importing classes: This method works for Java 6, 7 and 8
        var XDoc = com.xenos.framework.component.XDoc;

        //get the current JobTicket instance
        var jt = jobState.getJobTicket();

        //working with XDocs
        var xdoc = new XDoc("Hello World");
        jt.putInputMap("input", xdoc);

        var xdoc2 = jt.getInputXDocFor("input");
        var message = lib.getBytesAsString(xdoc2.toByteArray());
        logger.log("Value of input XDoc is " + message);

        //working with job variables
        jobState.setJobVar("foo", "bar");
        var foo = jobState.getJobVar("foo");
        logger.log("Value of foo is " + foo);
}
catch (exception) {
        // Catch any possible exception
        logger.log("Error in script: " + exception);
}
```

Figure 6-3: Sample usage of RunnableJobState

6.2.2 lib

The lib binding provides convenient library methods for scripts to utilize. Unlike job state objects, there is only one library object available, which is an instance of the `com.xenos.framework.script.JavaScriptLib` object.

A significant feature of JavaScriptLib is the ability to import existing library scripts into new scripts. This is intended to allow users to build up a library of functions that they wish to reuse throughout a series of scripts.

```
// Load an external JavaScript file.  If this file contains functions, then code below this
// will be able to call those functions.  If the file contains JavaScript outside of
// functions, then that code will be executed.  All variables defined in that code would
// be available below.
lib.load("${projectDir}counter.js");

// Call the getIncrementedCounter() method from counter.js
var counter = getIncrementedCounter();

// For the first iteration, call lib.listBindings() to tell us all about the
// binding objects available to the script.  This is akin to a printHelp() function
if (counter == 1) {
        listOfBindingObjects = lib.listBindings(false);
        listOfBindingObjectsWithMethods = lib.listBindings(true);

        jobState.getLogger().log("Available bindings:\n"+ listOfBindingObjects);
}

// Load a properties file into an object
var props = lib.loadProperties("${projectDir}configFiles/config.properties");
var username = props.getProperty("username");

// Add all of these properties to my job variables
jobState.getJobVars().putAll(props);

// Access a input stream
var inputStream = lib.getFileInputStream("${projectDir}rawData.bin");

// Load a text file into a String using the default platform encoding
var textFile = lib.getFileAsString("${projectDir}customers.txt");
var fileByteArray = lib.getFileAsBytes("${projectDir}logo.jpg");

// Use lib to create Java arrays that are Nashorn (Java 8) and Rhino (Java 6 & 7) compatible
var byteArray = lib.newByteArray(100);
var objectArray = lib.newObjectArray(100);
var booleanArray = lib.newBooleanArray(100);

try {
        // code that throws an exception
}
catch (exception) {
        // Use lib to get info about the exception, including message and line number
        var details = lib.getExceptionDetails(exception);

        // Log various different types of messages, which will alter return codes
        lib.logWarning("This is a warning, increases return code to 4");
        lib.logWarning("This is a warning, increases return code to 4, including exception details", exception);
        lib.logError("This is an error, increases return code to 8");
        lib.logError("This is an error, increases return code to 8, including exception details", exception);
}
```

Figure 6-4: Sample usage of JavaScriptLib

```
function getIncrementedCounter() {

        // Get the existing variable called "counter".  The second argument of "0" (zero) is the
        // default value that will be set if the job variable does not yet exist.  So the first
        // time this function is called during a job, counter will be initialized to 0.
        // Subsequent calls will retrieve the incremented value.
        var counter = jobState.getJobVarObject("counter", 0);
        counter++;

        // We have to overwrite the existing JobVar because "counter" is a stored as a
        // java.lang.Integer, which is immutable.   This technique would be required for all
        // other primitive types like byte, char, short, long and also String.  When modifying
        // properties of a more complex Object the setJobVar() may not be required.
        jobState.setJobVar("counter", counter);
        return counter;
}
```

Figure 6-5: Contents of counter.js

6.2.3 result

The result binding is utilized in parameter scripts as a mechanism to return the desired value of the parameter. The value of the result is treated as a string and then converted to the native type defined for the parameter (i.e. boolean, int, float, etc.). Any library scripts that get called during lib.load() will also have the ability to update the result binding.

Note: The result binding is ignored in component scripting.

```
// Set a default result value first
result = false; // could use "false" too

//Get a logger instance
var logger = jobState.getLogger();

try {
        if (jobState.getJobVar("EnableBooleanParameter") == "true") {
                // Update the result
                result = true;
        }
}
catch (exception) {
        logger.log("Error in script: " + exception);
}

// Let the script end, the script engine will take the final value of the result binding
```

Figure 6-6: Sample usage of result binding

6.3 Exception Handling and Default Job States

Exception handling is demonstrated in the sample code shown throughout the JavaScript component chapter by means of `try` and `catch` statements. While the usage of these is optional, they are highly recommended because they give you full control over your own error handling logic.

When a `try`/`catch` statement is not present, any exceptions encountered in the script are caught by the script executor. The effect of this depends on the context; in a script component, the job is aborted with an error while in parameter scripts, the error handling is less predictable because parameters are evaluated in hundreds of situations through the OpenText Output Transformation Suite engine where error handling logic varies. Parameter scripts also have to consider that it is technically

possible for parameter values to be requested when there is no active job or some other expected state objects not being populated. For instance, some parameters may be called during or prior to job initialization. In these situations, the jobState binding is available, but it may be a default job state object. The logger in the default job stats will log to the console (`standard out`). When using `try/catch` statements, you can anticipate the use of default job states and react accordingly.

6.4 JavaScript Compatibility

JavaScript support in OpenText Output Transformation Suite utilizes the JVM's built-in JavaScript script engine. This means the underlying JavaScript engine depends on the actual JVM being used at runtime.

Native JavaScript support was first introduced in Java 6 with JSR-223. This implementation used a modified version of Mozilla Rhino 1.6R2, which supports the JavaScript 1.5/ECMA-262 Edition 3 specification.

In Java 8, the Rhino engine has been replaced with Nashorn (the German word for Rhino). Nashorn provides much better performance and supports newer versions of JavaScript and ECMA scripts. It is important to be aware that there are some compatibility differences between Rhino and Nashorn so some scripts that run on Rhino may not necessarily run on Nashorn and vice versa.

Fortunately, script writers can be aware of the incompatibilities and consciously write scripts that are compatible with both engine types. This is especially important for users building applications that will run on Java 6 or 7 today, but may be upgraded to Java 8+ in the future. For more information, see "Creating Rhino and Nashorn Compatible Scripts" on page 160.

Java Version	Script Engine	JavaScript/ECMA Script Version
Default	`com.xenos.framework.util.jobstate.JobState`	The base job state. Provides access to properties such as job variables, job ID, active logger, and I/O utilities.
Output Transformation Designer	`com.xenos.framework.component.RunnableJobState`	In addition to the properties provided by the default job state, this Java class offers access to the JobTicket object for the running process.

6.4.1 Creating Rhino and Nashorn Compatible Scripts

To help maintain compatibility between Rhino and Nashorn scripts, follow these basic tips:

- Avoid using Rhino's JavaAdapter
- Avoid the use of importPackage and importClass
- Use Java native types instead of JavaScript types
- Use the provided JavaScriptLib (lib binding) to create arrays, ex: `lib.newByteArray(size)`
- Inspect JavaScript exceptions caught in try/catch blocks using `lib.getExceptionDetails(jsException)`

For further information on migrating from Rhino to Nashorn and highlights of the technical incompatibilities, see:

https://wiki.openjdk.java.net/display/Nashorn/Rhino+Migration+Guide (https://wiki.openjdk.java.net/display/Nashorn/Rhino%2bMigration%2bGuide)

Supplementary information about these JavaScript specifications can be found through the following links:

Java Specification	Link
Oracle Nashorn	http://www.oracle.com/technetwork/articles/java/jf14-nashorn-2126515.html
Mozilla JavaScript reference	https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
ECMA script specification	http://es5.github.io/

Chapter 7

Using SSL communication for OpenText services

There are a few steps involved in facilitating Secure Sockets Layer (SSL) communication for OpenText Java APIs and webservices. The topics in this section describe each step.

7.1 Step 1: Generating valid keystore and truststore certificate files

Before you can connect to the server, you must possess valid certificate files in order to authenticate your identity for access to the server. If you require a joint secure connection, then you must have a mutual certificate in both the server and client's keystores and truststores.

If you already have the certificate file(s) required, then skip to "Step 2: Enabling SSL protocol for OpenText Output Transformation Designer" on page 162.

If you need to generate the certificate file(s), you can create them using third-party applications such as Java KeyTools or OpenSSL. There are various methods of producing suitable certificate files, such as the sample method shown below:

```
>keytool -genkey -alias server -keyalg RSA -keystore server.keystore -dname
"CN=localhost, OU=MYOU, O=MYORG, L=MYCITY, ST=MYSTATE, C=MY"

>keytool -genkey -alias client -keyalg RSA -keystore client.keystore -dname
"CN=localhost, OU=MYOU, O=MYORG, L=MYCITY, ST=MYSTATE, C=MY"

>keytool -export -alias server -keystore server.keystore  -file server_pub.
key

>keytool -export -alias client -keystore client.keystore  -file client_pub.
key

>keytool -import -alias client -keystore server.keystore  -file client_pub.
key

>keytool -import -alias server -keystore client.keystore  -file server_pub.
key

>keytool -import -alias client -keystore client.truststore  -file client_
pub.key

>keytool -import -alias server -keystore server.truststore  -file server_
pub.key
```

7.2 Step 2: Enabling SSL protocol for OpenText Output Transformation Designer

Next, you must enable SSL protocols for Output Transformation Designer.

1. Go to `<OTS_Home>\settings` and open the `startup.properties` file for editing.
2. In the file, locate the `jvmargs` property, which appears similar to the following:

```
designer.jvmargs=-Xmx800M -Xms300M
designer.version=23.4.99_9999
designer.name=designer
```

3. You must add your truststore properties to the `jvmargs` value in the file. Update the `jvmargs` value with the following truststore information according to your environment:

```
designer.jvmargs=-Xmx800M -Xms300M  -
Djavax.net.ssl.trustStore="<truststore.jks_File_Path>" -
Djavax.net.ssl.trustStorePassword="<truststore_Password>"
designer.version=23.4.12_3456
designer.name=designer
```

   where

   <truststore.jks_File_Path> is the file path to the truststore.jks file.

   <truststore_Password> is the password required to access the truststore.

4. Save the file.

Note: Be aware that this is a Java properties file so the proper Java syntax must be respected; all backslash (\) characters must be delimited to be read properly. When setting file paths that use the backslash (\), they must be set using a `C:\\SampleFolder\truststore.jks` style. Alternatively, you can use a forward slash (/) in the file paths instead.

7.3 Step 3: Configuring your application server

In Output Transformation Designer, you must create an EAR file or deployment package using the **Package and Deploy Wizard** and indicate that the connection is a secure one. On the **Select the Target Application Server** screen of the Package and Deploy Wizard, select the **Is Secure** check box and in the **HTTP Port** and **JNDI Port** boxes, ensure that the port number you are using is for a secure port. Depending on your application server type, additional configuration steps may need to be performed on your application server; consult your application server's documentation for more details on enabling SSL communication.

Note: For more information on the Package and Deploy Wizard, see *OpenText Output Transformation Server - User Guide (VDTOTS-H-UGD)*.

Setting up SSL on Tomcat

To set up SSL on your Tomcat application server, you must complete the following tasks:

1. Creating a deployment package
2. Modifying the Tomcat configuration
3. Adding JVM parameters to the truststore

Before you begin, you must meet the following requirements:

- A valid keystore with a valid certificate for the Output Transformation Server instance
- A valid truststore with a public certificate

Creating a deployment package

You can create a deployment package using the Package and Deploy Wizard or by running the `SetupDeployment.bat/.sh` scripts.

When creating the deployment package, you must make sure that the fully qualified host name is used when specifying the host name and that you opt to use a secured http connection (in the Package and Deploy Wizard, select the **Is Secure** check box).

Modifying the Tomcat configuration

You must update the Tomcat configuration file with the required security and port settings.

1. Go to the `<OTS_home>/TomcatBase/<instance_Name>/conf` folder and open the `server.xml` file for editing.
2. Locate the `<Connector>` tag section of the file. It has a similar appearance to the following:

```
<Connector port="${ots.port}"
protocol="org.apache.coyote.http11.Http11NioProtocol"     maxThreads="150"
SSLEnabled="true" scheme="https" secure="true"      clientAuth="false"
sslProtocol="TLS"
keystoreFile="<keystore_Folder>\keystore.jks"
     keystorePass="<keystore_Password>"  />
```

   where

   <keystore_Folder> is the folder path location for the keystore file.

   <keystore_Password> is the password used to access the keystore.

3. Comment out the existing `<Connector>` tag for HTTP to turn off connecting through HTTP.
4. Uncomment the `<Connector>` tag for SSL to turn on connecting through SSL.
5. In the `<Connector>` section for SSL, set `${ots.port}` as the value for **port**.
6. Set the values for `keystoreFile` and `keystorePass` according to your environment.

7. Save your changes.

Adding JVM parameters to the truststore

You must add the JVM parameters to direct calls to the truststore.

1. Go to the `<OTS_home>/TomcatBase/<instance_Name>/bin` folder and open the `setenv.bat` file for editing.
2. In the file, add the following lines below the existing contents:

```
set CATALINA_OPTS=%CATALINA_OPTS% -
Djavax.net.ssl.trustStore="<truststore.jks_File_Path>"
set CATALINA_OPTS=%CATALINA_OPTS% -
Djavax.net.ssl.trustStorePassword=<truststore_Password>
```

   where

   <truststore.jks_File_path> is the truststore file path location.

   <truststore_Password> is the password used to access the truststore.

3. Save your changes.