Oracle® AutoVue Document Print Services (DPS)

Security Guide

Release 20.1.1

E24105-01

September 2011

ORACLE®

Oracle AutoVue DPS Security Guide, Release 20.1.1

E24105-01

Contents

Preface

This documentation provides guidelines on how to secure the AutoVue Document Print Services (DPS), all of its components, and the communication between the different components.

Audience

This document is intended for Oracle partners and third-party developers (such as integrators) who want to complement their existing print server solutions by leveraging AutoVue's powerful printing capabilities within their broader enterprise applications.

Related Documents

For more information, see the following documents:

- Oracle AutoVue DPS Deployment Guide
- Oracle AutoVue DPS Release Notes

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

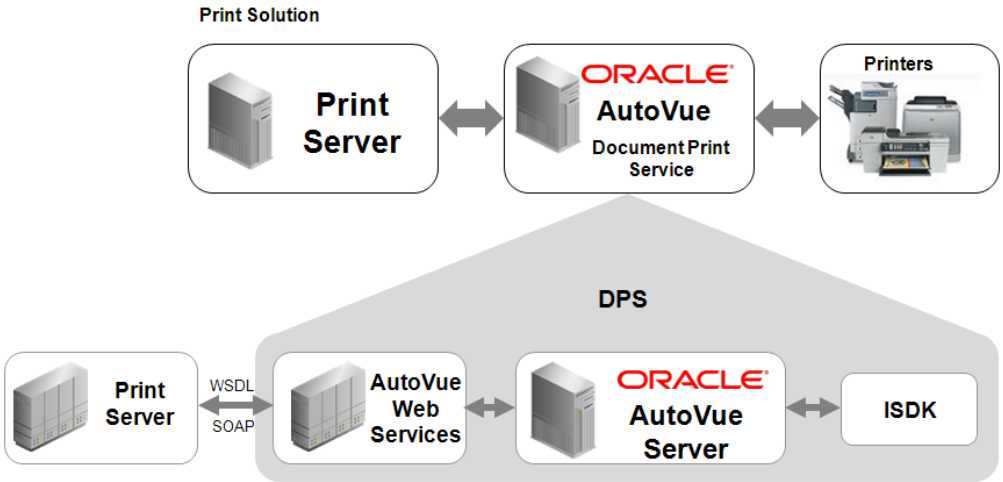
Overview

This section guidelines on how to secure the AutoVue Document Print Service (DPS), all of its components, and the communication between the different components.

Product Overview

AutoVue Document Print Services (DPS) is a reliable, scalable, secure product which allows organizations to complement their existing print server solutions by leveraging AutoVue's powerful printing capabilities within their broader enterprise applications.

DPS is a Web Service-based interface. The print server that consumes the DPS can be written in any programming language (for example, Java or any language in a .NET environment) as long as they understand Web Services Description Language (WSDL) and communicate using Simple Object Access Protocol (SOAP). The print server, DPS, and printers consist of a print solution as illustrated in the following figure.



In order to build a complete print solution, you must develop your own print server. The role of a print server may be as follows:

- Interact with your application to identify:
  - The files that need to be printed.
  - The printer to send each file.
  - The print options needed for each file. This includes paper size, page orientation, footers / headers, and so on.
- Collate print job requests.

- Call AutoVue DPS to perform actual printing.
- Poll printer for status.

Refer to the *AutoVue Document Print Services Deployment Guide*.

General Security Principles

This section outlines the general security principles of DPS.

Keep up to date on Software

One of the principles of good security practice is to keep all software versions and patches up-to-date. Throughout this document a DPS maintenance level of 20.1.1 or later is assumed.

Keep up to date on Latest Security Information

Oracle continually improves its software and documentation. Make sure you install the latest version of DPS.

Restrict Network Access to DPS

Keep the print solution which includes the print server, DPS, and printers behind a firewall. In addition, restrict access to DPS components deployed on Oracle Weblogic Server by leveraging a filtering mechanism provided by Weblogic application server.

Authentication

Allow a system to verify the identity of users that request access to DPS.

Data protection

Do not enable Web services SOAP message log as it may include sensitive information. Additionally, do not dump to the logs any sensitive username/password information.

Protect sensitive data against access by those who are not authorized users of the system. For example, an encryption mechanism may be used to protect data sent between the print server and DPS component.

2

Secure Installation and Configuration

This section describes the secure installation and configuration steps for the DPS application, and general guidelines on how to implement the print server.

Installation Overview

All the components included in the print solution should be installed in secure manner. The following sections cover the following topics.

- Installing Oracle Weblogic Application Server
- Installing Oracle AutoVue DPS
- Installing Oracle AutoVue Web Services
- Post Installation Configuration

Installing Oracle WebLogic Application Server

Follow the *Oracle Fusion Middleware Securing a Production Environment for Oracle WebLogic Server* document to install WebLogic Server in a secure manner and on a secure Weblogic server host.

The administrator should note the following items during the Weblogic installation and domain configuration:

- During installation of the WebLogic Server, select **Typical** type which includes all the necessary components that DPS applications require.
- During the process of configuring the Weblogic domain, select **Production Mode** instead of **Development Mode**.
- During the process of creating managed servers where DPS is deployed, you have the option to select the check box to enable the SSL, and specify SSL listen port.
- By default, WebLogic Server is configured with two keystores:
  - *DemoIdentity.jks*: Contains a demonstration private key for WebLogic Server. This keystore contains the identity for WebLogic Server.
  - *DemoTrust.jks*: Contains the trusted certificate authorities from the WL_HOME\server\lib\DemoTrust.jks and the JDK cacerts keystores. This keystore establishes trust for WebLogic Server.

These keystores are located in the WL_HOME\server\lib directory. For testing and development purposes, the keystore configuration is complete. Oracle highly recommends to not use the demonstration keystores in a production environment.

For information on configuring custom trust and identity keystores, see "Configuring Identity and Trust" in *Oracle Fusion Middleware Securing Oracle WebLogic Server.*

> **Note:** When you use the keytool utility, the default key pair generation algorithm is Digital Signature Algorithm (DSA). The WebLogic server does not support DSA. As a result, you must specify another key pair generation and signature algorithm when using WebLogic Server. Additionally, Oracle requires a key length of 1024 bits or larger.

DPS also relies on the following security services provided by Weblogic server.

- Associating Web services policy at runtime using Admin console.
- Using default Web services security configuration whose name is default_wss.
- Using connection filters to restrict access to AutoVue Web Services based on IP addresses and ports.

Later sections of this document illustrate how to configure the above mentioned security services in more detail.

Installing the DPS Application

The DPS application includes two components:

- AutoVue Server
- AutoVue Web Services

For information on how to securely install the AutoVue server, refer to the *Oracle AutoVue Secure Installation and Configuration Guide*.

For information on how to install AutoVue Web Services, refer to the *Oracle AutoVue Web Services Installation Guide*.

Post Installation Configuration

Security plays an important role in communication between applications. When it comes to Web services, this issue is even more critical. As a result, it is highly recommended for the print server to communicate with AutoVue Web Services using HTTPS protocol only.

Perform the following steps in order to run and use AutoVue Web Services over SSL.

1. Deploy AutoVue Web Services on a secure server. For deployment information, refer to "Appendix A" in the *Oracle AutoVue Web Services Installation and Configuration Manual*.
2. Export the certificate from Weblogic server into a file (it can be done through a Web browser).
3. Import the Weblogic server certificate to your client environment.
4. Generate and use the client proxy. Fore more information, refer to the "Importing and Using Client Proxy" section in the *Oracle AutoVue Web Services Developer's Guide*.

Note: Make sure you provide the HTTPS address of the WSDL (for example, https://host:port/AutoVueWS/VueBeanWS?wsdl).

DPS AutoVue Web Services consists of two Web applications: AutoVue Web Services and Vueservlet. The following sections discuss how to configure a secure communication between AutoVue Web Services and VueServlet via SSL.

Configuring AutoVue Web Services Application

- Open the AutoVue Web Services web.xml file in a text editor.
- Locate the value for the environment entry name ***initialJVueServer***.
- Change the protocol from **http** to **https**.
- Change port to SSL listen port.
- Save the changes and redeploy the AutoVue Web Services application.

Configuring VueServlet Application

- Open the VueServlet web.xml file in a text editor.
- Add the following init-param for VueServlet:

```
<init-param>
  <param-name>EnableSSL</param-name>
  <param-value>true</param-value>
</init-param>
```

- Save the changes and redeploy the VueServlet application.

For information on enabling secure communication (SSL) between the AutoVue Server and VueServlet application of DPS product, refer to the *Oracle AutoVue Secure Installation and Configuration Guide*.

If the DPS application processes the documents stored inside a document management system (DMS), the communication between the AutoVue server and the VueLink should be secured via SSL. Refer to individual VueLink installation guide for more information.

3

Security Features

This section outlines specific security mechanisms offered by the DPS application.

The Security Model

The DPS security requirements arise from the need to protect data from deliberate unauthorized attempts to access the file.

The critical security features that provide these protections are:

- Restrict IP Access - Allow only the print server to be able to invoke the Web services of DPS application.
- Authentication - Ensure that only authenticated individuals get access to DPS.
- Encryption and Sign - Encrypt and sign SOAP message between the print server and AutoVue Web Services.
- Authorization - This is only for the documents stored inside a DMS.

Configuring and Using Restrict IP Access

DPS does not require public access, it only needs to be accessed by the print server and the AutoVue server. To prevent unauthorized access to the DPS, it is recommended to tighten the deployment and limit access to AutoVue Web Services through a filtering mechanism provided by the Weblogic application server.

The following steps describe how to configure the filtering mechanism.

1. Log onto WebLogic Admin console.
2. From the left panel, select the domain that you want to configure (the domain that AutoVue Web Services is deployed on).
3. Select **Security** and then **Filter**.
4. Select the **Connection Logger Enabled** checkbox to enable the logging of accepted messages. The Connection Logger logs successful connections and connection data in the server. This information can be used to debug problems relating to server connections.
5. In the **Connection Filter** field, specify the connection filter class to be used in the domain.

   To configure the default connection, specify

   ```
   weblogic.security.net.ConnectionFilterImpl
   ```

6. In the **Connection Filter Rules** field, enter the syntax for the connection filter rules. The syntax is as follows:

   ```
   Target localAddress localPort action protocols
   ```

   The following is the recommended rule set (assuming that AutoVue Web Services is deployed on port 7011 and VueServlet is deployed on port 7012):

   ```
   # Allow access from the Weblogic application server machine
   <Weblogic IP or hostname> * * allow
   # Allow access from the AutoVue machine
   <autovue IP or hostname> * 7012 allow https
   # Allow access from the Print Server machine
   <Print Server IP or hostname> * 7011 allow https
   # Refuse the other access for all other machines
   <IP range to be restricted> * * deny
   ```

   Replace the <*Weblogic IP or hostname*> and <*autovue IP or hostname*> with the actual hostname or IP address of the machines.

   Replace the <*IP range to be restricted* > with the range of IPs that should be prohibited from accessing DPS AutoVue Web Services.

   For information on connection filter rules and syntax, refer to the "Using Network Connection Filters" section in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help*.

7. Click **Save**.

8. Restart the WebLogic Server so that your changes can take effect.

Note: If you accidentally enter rules that completely block access to the WebLogic server, and are no longer able to access the admin console, you must locate the config.xml file inside the WebLogic server machine (under the domain directory) and remove the <connection-filter-rule> parameters that deny access to the server from legitimate machines.

Configuring and Using Authentication, Encryption and Signature

Weblogic application server provides a mechanism to add WS-Security policy at runtime via Admin console. DPS leverages this mechanism to prevent unauthorized access. Refer to the `Securing WebLogic Web Services` document for more information.

Weblogic application server supports two security types for JAX-WS:

- *Transport-level security:* SSL is used to secure the connection between WS client and WS provider, may also include username/password for authentication.
- *Message-level security:* Data in a SOAP message is digitally signed or encrypted. May also include identity tokens for authentication.

Weblogic server bundles pre-packaged policy files. If the bundled policy files do not meet the customer's requirement, the Weblogic server also provides the mechanics to use the customer's own policy file. Users can associate pre-packaged policy files to Web Services at class-level, and also associate policies to a particular Web Services operation. The following sections provide examples on how to configure AutoVue Web Services to be secured by Weblogic pre-packaged policy.

Note: After you deploy DPS successfully on the Weblogic server continue the following examples.

Example for Securing DPS through Transport-level Policy

This example demonstrates how to associate pre-packaged policy "*policy:Wssp1.2-2007-Https-UsernameToken-Plain.xml*" to DPS AutoVue Web Services at class-level. This policy is a one-way SSL with plain text Username token. Client-side code should be updated in order to consume Web Services.

- Associate the policy file to DPS AutoVue Web Services application at class-level.

  Refer to information on attaching WS-policy file to a Web Service in the "*Configure a Policy File for a Web Service*" section in *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help* for detailed steps.

  Refer to Appendix B for DPS cluster deployment.

  Each time you update the policy files associating to AutoVue Web Services, you must update the DPS AutoVue Web Services deployment.

- Update Print Server-side code to invoke the Policy-Secured AutoVue Web Services

The Print Server-side code must provide the username and password to AutoVue Web Services. One new class *UserNameTokenHandler* is created. It implements *SOAPHandler<SOAPMessageContext>* and is used for putting user credential to SOAP header. The main method in Print Server uses *UserNameTokenHandler* to construct SOAP message to DPS AutoVue Web Services. The security-specific code in the sample client code is shown in bold.

```
import com.oracle.autovue.services.*;

import java.util.List;

import javax.xml.ws.BindingProvider;
import javax.xml.ws.handler.Handler;
import javax.xml.ws.handler.soap.SOAPHandler;
import javax.xml.ws.handler.soap.SOAPMessageContext;
import javax.xml.ws.soap.SOAPBinding;

public class PrintClient {

public static void main(String[] args) throws Exception {

   try{
      //create Service
      VueBeanWS_Service service = new VueBeanWS_Service();

      //create proxy
      VueBeanWS proxy = service.getVueBeanWSPort();

      //username or password for the UsernameToken
      String username = args[0];
      String password = args[1];

      BindingProvider bp = (BindingProvider) proxy;
      SOAPBinding binding = (SOAPBinding) bp.getBinding();

      // Add client side handlers via JAX-WS API
      List<Handler> handlerList =
```

```
      ((BindingProvider)proxy).getBinding().getHandlerChain();

      //username token
      UserNameTokenHandler handler = new UserNameTokenHandler();
      handler.setCredential(username, password);
      handlerList.add((SOAPHandler<SOAPMessageContext>)handler);
((BindingProvider)proxy).getBinding().setHandlerChain(handlerList);

      String URI = "server://@1/Samples/PADS_ILEARN.pcb;
      proxy.print(URI, null, null, null, null, false);

   }catch(Exception e){
      e.printStackTrace();

   }
  }
}
```

The following code defines the *UserNameTokenHandler* class.

```
import java.util.HashSet;
import java.util.Set;
import javax.xml.ws.handler.soap.SOAPHandler;
import javax.xml.ws.handler.soap.SOAPMessageContext;
import javax.xml.namespace.QName;
import javax.xml.soap.SOAPElement;
import javax.xml.soap.SOAPEnvelope;
import javax.xml.soap.SOAPHeader;
import javax.xml.ws.handler.MessageContext;


public class UserNameTokenHandler implements SOAPHandler<SOAPMessageContext> {

   private final String WSSE_NAMESPACE =  "http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd";
   private final String WSU_NAMESPACE = "http://docs.oasis-open.org/wss/2004/01/
    oasis-200401-wss-wssecurity-utility-1.0.xsd";
   private String user;
   private String pass;

   public void setCredential(String username, String pw){
    user = username;
    pass = pw;
   }

   @Override
   public Set<QName> getHeaders() {
     System.out.println("WSHandler: getHeader..");
     Set<QName> qnames = new HashSet<QName>();
     qnames.add(new QName ( WSSE_NAMESPACE,"Security","wsse"));
     return qnames;
   }

   @Override
   public void close(MessageContext context) {
      System.out.println("WSHandler close() called");
   }

   @Override
   public boolean handleFault(SOAPMessageContext context) {
      System.out.println("WSHandler handleFault() called");
```

```
      return true;
   }


   @Override
   public boolean handleMessage(SOAPMessageContext context)  {
      System.out.println("UserNameTokenHandler handleMessage() called");
      Boolean outboundProperty = (Boolean) context.get (MessageContext.MESSAGE_
        OUTBOUND_PROPERTY);
      if (outboundProperty.booleanValue()) {
        System.out.println("\n Outbound message:");

         try {
            if (user != null) {
               SOAPEnvelope envelope =
                 context.getMessage().getSOAPPart().getEnvelope();
                SOAPHeader header = envelope.getHeader();
                if (header == null ) {
                   header = envelope.addHeader();
                }

               SOAPElement security = header.addChildElement("Security", "wsse",
                   WSSE_NAMESPACE);

               SOAPElement usernameToken =
                  security.addChildElement("UsernameToken", "wsse");
                usernameToken.addAttribute(new QName("xmlns:wsu"), WSU_
                     NAMESPACE);

               SOAPElement username =
                  usernameToken.addChildElement("Username", "wsse");
                  username.addTextNode(user);

                if (pass != null) {
                  SOAPElement password = usernameToken.addChildElement("Password",
                       "wsse");
                           password.addTextNode(pass);
                }
         }

        } catch (Exception e) {
            e.getStackTrace();
        }

        } else {
        System.out.println("\n Inbound message:");
        }

    return true;
  }
}
```

Example for Securing DPS through Message-level Policy

This example demonstrates how to associate the following pre-packaged policies to DPS AutoVue Web Services at class-level.

- Wssp1.2-Wss1.0-UsernameToken-Plain-X509-Basic256.xml
- Wssp1.2-EncryptBody.xml

- Wssp1.2-SignBody.xml

For a description of these policies, see *Oracle Fusion Middleware Securing WebLogic Web Services for Oracle WebLogic Server*.

After attaching these polices to DPS (similar steps as preceding section "Example for Securing DPS through Transport-level Policy"), the following steps must be performed.

1. Obtain two private key and digital certificate pairs to be used by the Web services runtime. One of the pairs is used for digitally signing the SOAP message and the other for encrypting it.

   Refer to the "Obtaining Private Keys, Digital Signatures, and Trusted Certificate Authorities" section in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

2. Load the private key and digital certificate pairs you obtained in the step 1 into the identity keystore.

3. Using the Administration Console, create the default Web service security configuration, which must be named **default_wss**.

   Refer to the "Create a Web Service Security Configuration" section in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help*.

4. Update the default Web services security configuration you created in the step 3 to use one of the private key and digital certificate pairs for digitally signing SOAP messages.

   Refer to the "Specify the key pair used to sign SOAP messages" section in *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help.*

5. Update the default Web services security configuration you created in the step 3 to use the second private key and digital certificate pair for encrypting SOAP messages.

   Refer to the "Specify the key pair used to encrypt SOAP messages" section in *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help*.

6. Configure Weblogic server to accept X.509 certificates.

   Refer to the "Use X.509 certificates to establish identity" section in *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Help*.

7. Create a keystore used by the client application. Similar as step 1. Ensure that WebLogic Server is able to validate the X.509 certificate that the client uses to digitally sign its SOAP request, and that WebLogic Server in turn uses to encrypt its SOAP responses to the client.

   Refer to the "SSL Certificate Validation" section in *Oracle Fusion Middleware Securing Oracle WebLogic Server.*

8. Update Print Server code which needs to import Weblogic Web Services security API to create needed client-side credential providers. Use the **ClientBSTCredentialProvider** WebLogic API to create a binary security token credential provider from the client's certificate, private key, etc. Use the **ClientUNTCredentialProvider** WebLogic API to create a username token from the client's username and password. The security-specific code in the sample client is shown in bold.

Note: The libraries under <MiddleWare_Home>\modules and <WL_Home>\server\lib should be in the build path of Print Server's project

---

```
import java.security.cert.X509Certificate;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import javax.xml.ws.BindingProvider;

import weblogic.security.SSL.TrustManager;
import weblogic.wsee.security.bst.ClientBSTCredentialProvider;
import weblogic.wsee.security.unt.ClientUNTCredentialProvider;
import weblogic.wsee.security.util.CertUtils;
import weblogic.xml.crypto.wss.WSSecurityContext;
import weblogic.xml.crypto.wss.provider.CredentialProvider;

import com.oracle.autovue.services.VueBeanWS;
import com.oracle.autovue.services.VueBeanWS_Service;


public class Message_Level_Client {

   public static void main(String[] args) throws Exception {
      try{
         //Create Service
         VueBeanWS_Service service = new VueBeanWS_Service();

         //create proxy
         VueBeanWS proxy = service.getVueBeanWSPort();

        //username or password for the UsernameToken
        String username = args[0];
        String password = args[1];

        //client keystore file
        String clientKeyStore = args[2];
        String clientKeyStorePass = args[3];
        String clientKeyAlias = args[4];
        String clientKeyPass = args[5];

       //server certificate
       String serverCertFile1 = args[6];
       String serverCertFile2 = args[7];
       X509Certificate serverCert1 =
(X509Certificate)CertUtils.getCertificate(serverCertFile1);
       X509Certificate serverCert2 = (X509Certificate)
CertUtils.getCertificate(serverCertFile2);

       //create emtpy list of credential providers
       List credProviders = new ArrayList();

       //Create client-side BinarySecurityToken credential provider
       // X.509 for identity, based on certificate and keys parameters
       CredentialProvider cp = new ClientBSTCredentialProvider(clientKeyStore,
clientKeyStorePass, clientKeyAlias, clientKeyPass, "JKS", serverCert1);
       credProviders.add(cp);

      CredentialProvider cp0 = new ClientBSTCredentialProvider(clientKeyStore,
clientKeyStorePass, clientKeyAlias, clientKeyPass, "JKS", serverCert2);
       credProviders.add(cp0);

         CredentialProvider cp1 = new
```

```
            ClientUNTCredentialProvider(username.getBytes(),
                 password.getBytes());
           credProviders.add(cp1);

          // Set stub property to point to list of credential providers
          Map<String, Object> rc = ((BindingProvider)
             proxy).getRequestContext();
           rc.put(WSSecurityContext.CREDENTIAL_PROVIDER_LIST, credProviders);


        // setup the TrustManager.
        rc.put(WSSecurityContext.TRUST_MANAGER,
              new TrustManager() {
             public boolean certificateCallback(X509Certificate[] chain,
           int validateErr) {
                return true;
           }
    });


       String URI = "server://@1/Samples/PADS_ILEARN.pcb;
       System.out.println(proxy.print(URI,null,null,null,null,false));

  }catch(Exception e){
       System.out.println("caught "+e.toString());
       e.printStackTrace();
       System.out.println(e.getMessage());
    }
  }
}
```

Configuring and Using Authorization

This section only applies for DPS processing the documents stored inside a DMS.

DPS relies on the security feature provided by DMS. Print Server submits the request including the user's credential information to DPS, and DPS in turn passes the user credential information to the DMS. The DMS secures the document based on user access level to the document. DPS prints out the document if user has at least Read permission to the document.

Refer to "AutoVue Web Services and DMS Integrations" section in *Oracle AutoVue Web Services Developer's Guide* for information on configuring DPS to process the documents inside a repository.

Configuring and Using Other Security Features

AutoVue Server supports Upload, HTTP/HTTPS, FTP/FTPS, and Server protocol in the filename URL. By default, the upload protocol is disabled in the DPS for security reasons. From the performance point-of-view, it is also not recommended to use an upload protocol. However, if necessary, the upload protocol can be enabled by setting the **isUploadProtocolEnabled** inside **web.xml** of the AutoVueWS Web application to TRUE.

If you are enabling upload protocol, a printing client can print any file that is accessible to the AutoVueWS Web application. So, if not all users should have access or some should have partial access to those files, then the file access management should be addressed in the print service client code and only authorized print service client connection should be allowed to access AutoVue print service.

It is recommended to use a DMS for storing documents rather than an upload protocol, and to use an ISDK-based integration along with print services to access the documents by passing the user's credentials.

In case using a content repository and VueLink is not feasible, then choose either an FTP or Server protocol with the AutoVue and print service. Refer to AutoVue documentation for information on setting up Server protocol.

A

Secure Deployment Checklist

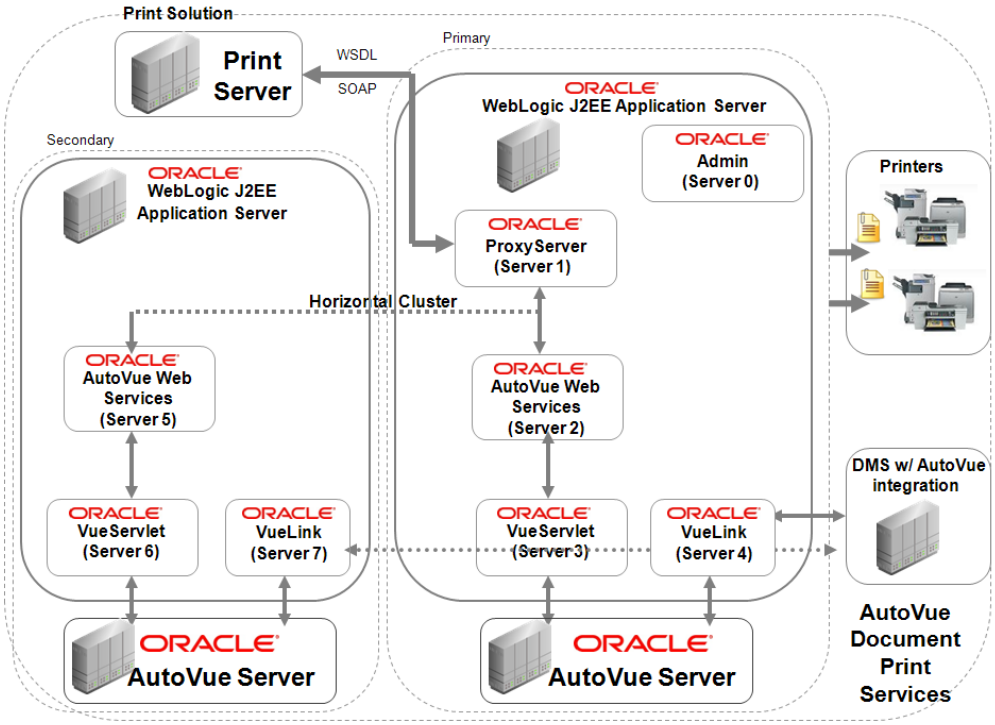
The section provides a secure deployment checklist.

- Install Print Server and DPS behind a firewall.
- Install Oracle Weblogic server in a secure manner.
- Install Oracle AutoVue Server in a secure manner. Refer to the *Oracle AutoVue, Client/Server Deployment Security Guide*.
- Enable SSL for all the managed servers that deploy DPS.
- Configure Connection filter to allow only the Print Server to access DPS AutoVue Web Services application.
- Associate the WS-Security policy to DPS AutoVue Web Services at runtime via Oracle Weblogic Admin console.
- Keep DPS software up-to-date.
- Keep up-to-date on latest security information.
- Keep the AutoVue server up-to-date.

B

Associating WS-Security Policies to DPS Applications

When associating WS-Security policies to DPS AutoVue Web Services applications deployed on two managed servers which are on two machines, the following steps must be followed.



Note: For machine deployment, see "Four Machine Deployment (Horizontal Cluster)" in *AutoVue Document Print Service Deployment Guide*.

1. Associating WS-Security policies to DPS AutoVue Web Services application deployed on the AutoVue Web Services (Server 1) on the primary machine (Machine-1) can be done via Weblogic Admin console.

Note: When you conduct the step you must associate policies to AutoVue Web Services, the assistant asks you for the directory that should contain the deployment plan. By default, it is *C:\Oracle\AutoVueWS\autovue_webservices\AutoVueWS\Plan.xml.* Also one **plan** folder is created automatically. The default location is under the directory *C:\Oracle\AutoVueWS\autovue_webservices*. Take note of these two locations, as they are used in step 2.

2. Associate WS-Security policies to DPS AutoVue Web Services application deployed on the AVWSServer-2 on Machine-2.

   - Copy the **Plan.xml** and also **plan** folder created in step 1 to the second machine. Keep them as the same path as on the machine -1. By default, they are located under *C:\Oracle\AutoVueWS\autovue_webservices\AutoVueWS\Plan.xml* and *C:\Oracle\AutoVueWS\autovue_webservices\plan.*
   - Update the AutoVue Web Services application deployed on Machine-2 by specifying the deployment plan path which is *C:\Oracle\AutoVueWS\autovue_webservices\AutoVueWS\Plan.xml* by default.

Note: Each time, after you modify the associated policy files to AutoVue Web Services, not only the DPS AutoVue Web Services deployed on Machine-1 needs to be updated, but also the one deployed on the Machine-2. Note that you only need to copy **Plan.xml** and **plan** folder to Machine-2 the first time. That means when you update the associated policy files later no copy action is required.

C

Feedback

If you have any questions or require support for AutoVue please contact your system administrator. Some customization and maintenance must be done on the server and cannot be implemented on the client machine. If the administrator is unable to resolve the issue, please contact Oracle Corp.

General Inquiries

Telephone	+1.514.905.8400 or +1.800.363.5805
E-mail	autovuesales_ww@oracle.com
Web Site	http://www.oracle.com/us/products/applications/autovue/index.html

Sales Inquiries

Telephone	+1.514.905.8400 or +1.800.363.5805
E-mail	autovuesales_ww@oracle.com

Customer Support

Web Site	http://www.oracle.com/support/index.html