



Web Services Administrator Guide

- SAP BusinessObjects Business Intelligence platform 4.0 Feature Pack 3

2012-03-14

Copyright

© 2011 SAP AG. All rights reserved. SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company. Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary. These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

2012-03-14

Contents

Chapter 1	Document History.....	5
Chapter 2	Getting Started.....	7
2.1	About this documentation.....	7
2.1.1	Who should use this documentation?.....	7
2.1.2	About the Web Services.....	7
2.2	Migrating your Web Services.....	9
2.3	Support for legacy consumer applications.....	9
Chapter 3	Configuring your deployment.....	11
3.1	Verifying your deployment.....	11
3.2	To configure which Web Services are active.....	12
3.3	Configuring the dsws.properties file.....	12
3.4	Deploying the Web Services with clustering.....	16
3.5	Deploying the Web Services using a DMZ.....	18
Chapter 4	Securing your deployment.....	19
4.1	Public-key encryption	19
4.2	Client and server certification	20
4.3	To make a secure connection using SSL.....	21
4.4	Generating certificates and keystores using JDK	22
4.5	To configure your Tomcat servlet container for SSL	23
4.6	Consuming the Web Services over SSL.....	24
Chapter 5	Scaling and performance	27
Appendix A	More Information.....	29
Index		31

Document History

The following table provides an overview of the most important document changes.

Version	Date	Description
SAP BusinessObjects Business Intelligence platform 4.0	November, 2011	First release of this document.

Getting Started

2.1 About this documentation

This documentation provides you with information and procedures for configuring and administering your SAP BusinessObjects Business Intelligence (BI) platform Web Services deployment. Procedures are provided for common tasks. Conceptual information and technical details are provided for all advanced topics.

For information on how to deploy BI platform web applications to a supported Java web application server, including the Web Services `dswsbobje.war` file, see the *SAP BusinessObjects Business Intelligence Platform Web Application Deployment Guide*.

For information about installing the BI platform, see the *SAP BusinessObjects Business Intelligence Platform Installation Guide*.

For information related to the administration of a BI platform server, see the *SAP BusinessObjects Business Intelligence Platform Administrator Guide*.

2.1.1 Who should use this documentation?

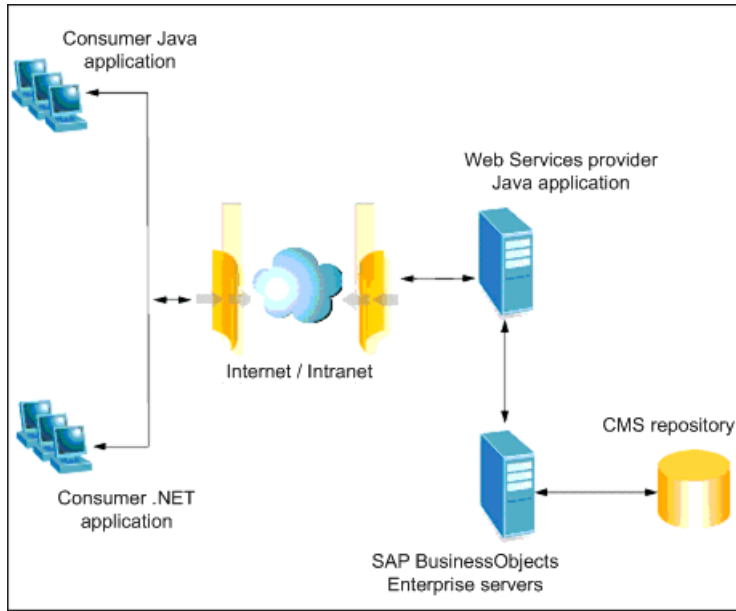
This help covers configuration tasks. We recommend consulting this guide if you are:

- Configuring your first deployment.
- Making significant changes to the architecture of an existing deployment.

This documentation is intended for system administrators who are responsible for configuring, managing, and maintaining the BI platform Web Services. Familiarity with your operating system and your network environment is beneficial, as is a general understanding of web application server management and scripting technologies. However, to assist all levels of administrative experience, this documentation aims to provide sufficient background and conceptual information to clarify all administrative tasks and features.

2.1.2 About the Web Services

The BI platform Web Services allow you remotely interact with objects in your BI platform deployment over HTTP. This makes the integration of your deployment with other web-based applications easier. The Web Services are deployed in two parts: the provider web application and a set of consumer APIs.



Provider application

The provider application (`dswsbobje.war`) is one of many deployed web applications within a BI platform system. The provider application makes the Web Services available to other client applications to interact with. The Web Services provider application follows the WS-Interoperability Basic Profile 1.0 and is implemented using Apache Axis2 1.3.

Note:

Since Apache Axis2 is used for the implementation of the Web Services, the resulting provider application is a Java web application which must be deployed on a Java web application server.

This documentation contains information on how to configure a deployed provider application (`dswsbobje.war`). To learn how to deploy the provider application, see the *SAP BusinessObjects Business Intelligence Platform Web Application Deployment Guide*.

Consumer APIs

Client applications that consume and use the web services exposed by providers are consumer applications. The BI platform offers a set of Java and .NET consumer APIs that allow developers to implement user authentication and security, document and report access, scheduling, publications, and server management within your installation. The Consumer APIs follow WS-Interoperability Basic Profile 1.0 and it is recommended that they be used when developing consumer applications with the BI platform Web Services.

To learn how to develop custom applications using the Consumer APIs, see the *SAP BusinessObjects Business Intelligence Platform Web Services Consumer SDK Developer Guide*. To learn how to install

the BI platform, which includes the consumer SDKs, see the *SAP BusinessObjects Business Intelligence Platform Installation Guide*.

2.2 Migrating your Web Services

This section contains information about changes to be aware of when you upgrade from an earlier version of the SAP BusinessObjects Business Intelligence platform Web Services.

Obsolete services

The `QueryService`, `SaveService`, and `ReportEngine` services are now obsolete and no longer supported. Use the `BIPlatform` service as a replacement.

Lower case service names are now deprecated

Lower case names to reference the Web Services, such as `session`, `biplatform`, `liveoffice`, `bicatalog`, and `publish`, are now deprecated. Since service names are case-sensitive, use the following supported services names within your application:

- `Session`
- `BIPlatform`
- `LiveOffice`
- `BICatalog` (deprecated in XI 3.0 - use the `BIPlatform` service as a replacement)
- `Publish` (deprecated in XI 3.0 - use the `BIPlatform` service as a replacement)

For example: `http://<servername>:<port>/dswsbobje/services/Session`

2.3 Support for legacy consumer applications

SAP BusinessObjects Business Intelligence platform 4.0 providers are backwards-compatible with the SAP BusinessObjects Enterprise XI 3.x version of consumer applications. 4.0 and XI 3.x providers are built on the Apache Axis2 web service framework. The BusinessObjects Enterprise XI Release 2 consumers were built on the Apache Axis 1.1 framework, and are not compatible with XI 4.0 or XI 3.x consumers.

Configuring your deployment

This section explains how you can verify and configure your Web Services deployment. Instructions are provided on how to validate an existing deployment of the Web Services provider application, and what configuration properties are available to customize the deployment to your needs.

3.1 Verifying your deployment

To verify that the provider application is deployed correctly, open the following URL in a web browser:

```
http://<servername>:<port>/dswsbobje
```

The "Axis2 - Home" page appears displaying a greeting and a link to the listServices page.

Note:

- Substitute `<servername>` and `<port>` with the machine name and port where the Web Services are deployed.
- If the "Axis2 - Home" page fails to appear or displays an error message, the provider application has failed to deploy. See the *SAP BusinessObjects Business Intelligence Platform Web Application Deployment Guide* to troubleshoot and redeploy.

WSDL and service verification

To consume the Web Services from a custom application, developers can use the Java or .NET consumer APIs that are provided with an installation of the BI platform. Developers may however opt to generate the API from the WSDLs that are exposed by the provider. The WSDLs follows the WS-Interoperability Basic Profile 1.0.

Note:

Only the Java and .NET consumer APIs provided by SAP are supported. APIs generated from the WSDLs are not officially supported. For more information on the WSDL and consumer APIs, see the *SAP BusinessObjects Business Intelligence Platform Web Services Consumer SDK Developer Guide*.

To verify that the WSDLs are deployed correctly and can be accessed, open the following URL in a web browser:

```
http://<servername>:<port>/dswsbobje/services/listServices
```

A list of all active Web Services appears. From here you can verify a few things:

- Click on the hyperlink for each named Web Service to view the WSDL for that service.

- In the WSDL, view the descriptions of the address bindings. Ensure that the correct server name, port number, and web application server name is in the binding URLs.

Redeploying `dswsbobje.war`

The Web Services provider application (`dswsbobje.war`) is automatically installed and deployed to your specified Java application server during a default installation of the BI platform. For information on how to redeploy the provider to a new or existing web application server, see the *SAP BusinessObjects Business Intelligence Platform Web Application Deployment Guide*. This documentation contains instructions on how to deploy all BI platform web applications, including `dswsbobje.war`. Topics include:

- How to deploy web applications using the WDeploy command-line or Windows-based GUI tools.
- How to manually deploy web applications using the administrative consoles of supported web application servers.
- Supported deployment scenarios.

3.2 To configure which Web Services are active

Each individual Web Service in your deployment can be activated or deactivated based on a property value in the service's `services.xml` file. There is a `services.xml` file for each Web Service. The following examples shows you how to deactivate the BICatalog service.

1. Locate and open the `services.xml` file for the service.

For example, in a default Windows installation, the file for the BICatalog service is located at `C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\warfiles\webapps\dswsbobje\WEB-INF\services\bicatalog\META-INF\`

2. Add an `activate` property to the `service` tag and set the value to `false`.

For example, if you are disabling BICatalog service, this is what your changes must look like:

```
<service name="BICatalog" activate="false">
```

3. Restart your web application server.

Note:

When you deactivate a service using the above procedure, the WSDL file of the service that you deactivated may still be accessible. However, this does not mean that the service is still active. If you try to invoke the service operation in your code, an error message appears. This message verifies that the service was turned off.

3.3 Configuring the `dsws.properties` file

This section shows how to configure the runtime behavior of your Web Services. The `dsws.properties` file contains configuration information that you can modify in your deployment. In a default Windows

installation, this file is located at `C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\warfiles\webapps\dswsbobje\WEB-INF\classes\`.

The following properties are defined in the `dsws.properties` file and can be configured:

Property	Description
domain	<p>The domain name of your Central Management Server (CMS). This property must be entered as <code>CMSname:CMSport</code>. To increase security, it is recommended that you configure the name of your CMS domain to ensure that the Web Services are able to access only the proper CMS. If the domain is not configured, malicious users may be able to access other CMS servers on your network.</p> <p>Note: This property must be set if you are configuring a clustered deployment where your Web Services are hosted on a separate server from the CMS. If this property is not correctly configured, applications dependent on the system's Web Services, such as SAP Crystal Reports for Enterprise 4.0, may fail to run properly.</p>
entry.size	<p>Specifies the maximum number of characters that can be stored in a dictionary entry. Setting <code>entry.size</code> to 0 or leaving it blank means database entries can be of any size.</p>
temp.dir	<p>The directory where temporary files are stored.</p>
image.dir	<p>The directory in which the temporary image files will be stored when retrieving a Crystal report in HTML format.</p>
image.cleanup	<p>Set this property to <code>true</code> to have the temporary images deleted after the image content is fetched. If this property is not set to <code>true</code>, an image cleaner background thread will run and periodically delete images in the directory specified by the <code>image.dir</code> property.</p>

Property	Description
report.server	<p>The report server to use with SAP BusinessObjects Web Services.</p> <p>The default setting is <code>ps</code>.</p>
bicatalog.defaultObjectProperties=si_instance,si_name,si_description	<p>The default properties for the BICatalog Web Service.</p>
clustered	<p>Set this property to <code>true</code> to enable clustering.</p> <p>The default setting is <code>false</code>.</p>
fileCachePath	<p>The directory where file download and upload cache data is stored.</p> <p>The default value is <code>c:\Program Files\Business Objects\Attachment File Cache</code>.</p>
checksumMethod	<p>Specifies the checksum verification method. This is used to verify the integrity of the file chunks being downloaded.</p> <p>Possible values:</p> <p><code>NONE</code>. No verification performed.</p> <p><code>BYTECOUNT</code>. Verifies based on the length, or byte count, of the file chunk.</p> <p><code>MD5</code>. Verifies based on MD5, a sophisticated fingerprint algorithm.</p> <p><code>SIMPLE</code>. Verifies by calculating the sum of each byte in a file chunk, and uses this sum for comparison.</p> <p>The default setting is <code>SIMPLE</code>.</p>
maximumChunkSize	<p>Specifies the maximum size of a chunk, in bytes, that can be uploaded with a single <code>BIPlatform.uploadFile</code> method call.</p> <p>The default value is <code>262144</code>.</p>

Property	Description
maximumUploadFileSize	<p>Specifies the maximum size of a file, in bytes, that can be uploaded.</p> <p>The default value is 10485760.</p>
numChunksPerSDKChunk	<p>Specifies the number of 64 kilobyte chunks that are downloaded with a single <code>BIPlatform.downloadFile</code> method call. For example, if this value is set to 2, then each call downloads 128 kilobytes of data.</p> <p>The default value is 4.</p>
defaultPageSize	<p>Specifies the number of InfoObjects returned in a single page. Pages are returned by calling the <code>BIPlatform.get</code> method with a path query.</p> <p>The default value is 100.</p>
noPagingQueryLimit	<p>Specifies the maximum number of InfoObjects returned by a path query if paging is not used. To page your query results, you must ensure that each InfoObject property in the <code>OrderBy</code> clause is an indexed property or is specified in the <code>allowedOrderBy</code> property.</p> <p>The default value is 1000.</p>
defaultSelectList	<p>Specifies a comma-delimited list of query language properties returned by default with each path query. Use the asterisk character (*) to specify all properties.</p>
searchSelectList	<p>Specifies a comma-delimited list of query language properties returned by search. Use the asterisk character (*) to specify all properties.</p>
allowedOrderBy	<p>Specifies a comma-delimited list of query language properties that you want to order by. This property allows you to page and sort by query language properties that are not indexed.</p>

Property	Description
validateXML	Specifies whether to validate each XML message passed through the Web Services provider. Possible values are <code>true</code> and <code>false</code> . The default value is <code>false</code> .

Example: `dsws.properties` file example

The following is an example of the `dsws.properties` file:

```
Trace=0
entry.size=255
temp.dir=
image.dir=
image.cleanup=false
domain=MyCMS:6400
report.server=ps
bicatalog.defaultObjectProperties=si_instance,si_name,si_description
```

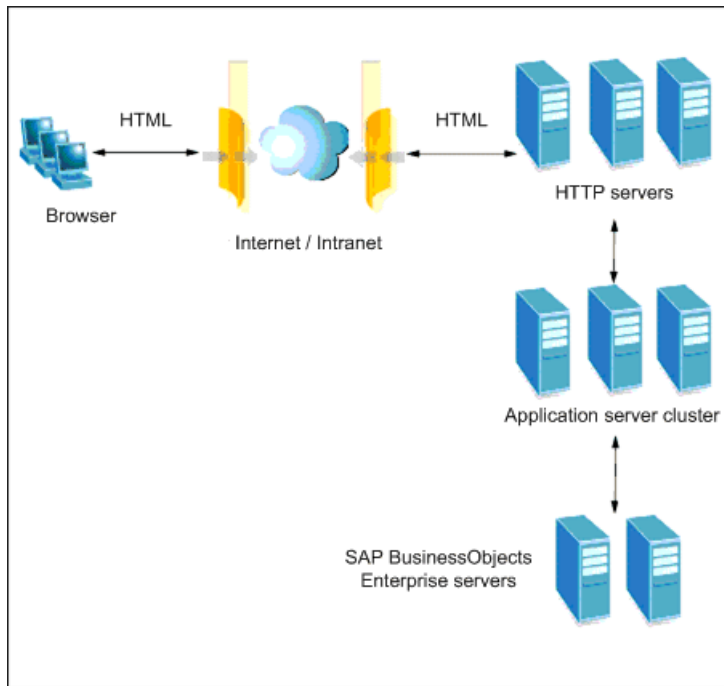
3.4 Deploying the Web Services with clustering

You can deploy the Web Services in a clustered configuration. Deploying the Web Services with clustering enabled provides greater scalability and failover.

Note:

Clustering is sometimes also referred to as Web Farms.

The following diagram illustrates a typical clustered deployment:



Enabling clustering in your Web Services provider deployment

To enable your provider applications to operate in a clustered environment, open the `dsws.properties` file on each server and change the value of the `clustered` parameter to `true`

Next, configure an images directory that will be shared by all of the application servers in the cluster. Update the `tmp_images` property in the `dsws.properties` files on each server to point to the shared images directory.

Next, edit the `web.xml` file uncommenting the `<distributable/>` tag between the `<webapps>` tags as shown as follows:

```
<web-app>
<display-name>dsws2</display-name>
  <!-- when using tomcat webserver cluster, please make <distributable /> available by recommenting follow
  line -->
<distributable />
```

Note:

- The `web.xml` file is a part of the war file. The default location of the `web.xml` file is as follows:
`\Program Files\SAP Business Objects\Tomcat\webapps\dswsbobje\WEB-INF`
 Substitute your web application server name for `Tomcat`
- Refer to your application server documentation for information on how to set up your application servers with clustering enabled.

Developing consumer applications against clustered providers

To work with a provider of Web Services with clustering enabled, consumers of the Web Service must maintain their session state. If the consumer application uses the consumer API shipped with the SAP BusinessObjects Business Intelligence platform, this is done automatically. If the consumer application

does not use the provided consumer API (and uses a consumer generated from the WSDL), the session must be maintained manually.

The `ServerInfo` class is provided as part of the `Session` service so that the Web Services consumers can check if the provider is using a clustered deployment. The consumer should use this class to check if the provider is using a clustered deployment. If so, the consumer must maintain session state.

Using `ServerInfo` to check for a clustered deployment

The `Session.getServerInfo()` method is used to obtain a `ServerInfo` object. Once a `ServerInfo` object has been obtained, the `ServerInfo.getClustered()` (Java) and `ServerInfo.Clustered()` (.NET) methods are used to determine whether the provider is using a clustered deployment. These methods return the Boolean `true` if clustering is used and `false` otherwise. If clustering is used, the consumer must maintain session state.

Maintaining session state

Refer to the MSDN online documentation for information on how to maintain session state for Web Services using .NET APIs. Session state is maintained by default when using the Java consumer API.

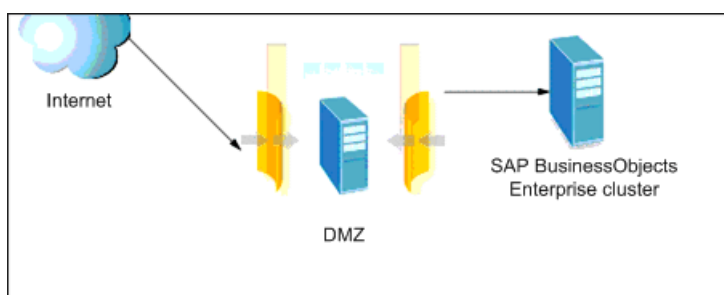
3.5 Deploying the Web Services using a DMZ

Because the BI platform Web Services employs extranet level security, you could deploy your Web Service using a DMZ to protect information.

The term demilitarized zone, or DMZ, describes a network topology where the application server is behind a firewall, and separated from the web server, which runs on a different subnet. The web server is then protected by a firewall from the external network.

The DMZ is thus a protected buffer zone between an organization's intranet and the external network or Internet. It is designed to keep outside users from accessing a server containing company data.

An illustration of this is shown below.



Refer to your *SAP BusinessObjects Business Intelligence Platform Administrator Guide* for further information on how to install a DMZ deployment.

Securing your deployment

To ensure the security of Web Services data transfers, use Secure Sockets Layer (SSL) to encrypt all SAP BusinessObjects Web Services responses. SSL encrypts data as it passes between a Web Service and its consumer. It also provides transaction validation because it prevents data from being modified while en route from sender to receiver. Changes in the data invalidates its authentication code and alerts the receiver that the data has been modified. Therefore, the two major benefits of SSL over most other encryption or data protection schemes are as follows:

- Public-key encryption exchanges data without transmitting data that is unsecure or data that would permit snoopers to decrypt the transmissions.
- Client and server authentication is accurate, because authentication certificates permit positive identification of the server, and optionally of the client, by exchanging third-party validated certificates.

4.1 Public-key encryption

There are two primary methods of encrypting data:

- Symmetric-key cryptography
- Public-key cryptography

In the context of how web services are used, the key is a random number, one that is usually quite large and complicated. Encryption engines take the key and the data, pass the key to the encryption algorithm, and process the data accordingly.

Symmetric-key cryptography is the simpler methods. It uses the same key both to encrypt and decrypt the data. The sender gives the data and the key to the encryption engine and passes the resulting encrypted data to the receiver. The receiver passes the encrypted data and key to the decryption engine.

However, symmetric-key cryptography introduces a security risk. The key used to encrypt and decrypt the data could be intercepted and used to decrypt (and view or disrupt) any later communications that use that key.

Public-key cryptography solves this problem by using two keys, a public key to encrypt data and private key to decrypt data:

- The public key is available to anyone who wants to send encrypted messages or data.
- The private key which remains private is never sent over the network, is used to decrypt data encrypted that has been encrypted with the public key.

Data encrypted with the public key can only be decrypted with a private key. The process of public-key encryption works as follows:

- Client A begins a secure session with Server B. A requests B's public key.
- B sends its public key to A.
- A receives B's public key, encrypts a message using the key, and sends the message to B.
- B receives the encrypted message and decrypts it using its private key.

Most encryption schemes, including SSL, use a combination of symmetric- and public-key encryption. Public-key encryption is used to actually establish the secure connection where a randomly generated symmetric key can be exchanged. The rest of the transaction encrypted by the quicker symmetric-key encryption method.

4.2 Client and server certification

For positive identification of servers and, optionally, clients SSL uses digital certificates. Digital certificates provide a means of positive identification over the Internet. The certificates that SSL uses are defined in the X.509 standard issued by the International Telecommunications Union (ITU). The items contained in an X.509-compliant digital certificate include:

- Name of the certificate owner
- Name of the certificate issuer
- Public key used by the owner
- Signature of the Issuer
- Valid dates

SSL provides three different levels of security, each based on certificates.

Level	Requirements
No security	No certificates required. No encryption applied to any transactions.

Level	Requirements
Server certification	<p>A server passes a certificate to a client. The client that verifies that the certificate is still valid and that it was issued by a trusted authority (trusted authorities are third-party issuers of digital certificates). You usually need to configure your HTTP server to recognize a certificate issuer as a trusted authority. Refer to your HTTP server administration manuals.</p> <p>If the certificate is valid and from a trusted issuer, the client uses the public key contained in the server certificate to begin the encrypted session.</p>
Client certification	<p>Both server and client exchange certificates. Both must check the certificate they receive for the validity date and the issuing authority. Like the server, browsers must be configured to specify which certificate issuers are trusted authorities.</p> <p>If both server and client accept the other's certificate, the session begins.</p> <p>You cannot use client certification to manage which users you want to grant access to your site. Although client certificates do contain the user's distinguished name, this is verified but not used to decide whether the user gets access to the site. Access is based solely on whether the certificate is valid and the issuer of the certificate is trusted by the server.</p>

You cannot use client certification to manage which users you want to grant access to your site. Although client certificates do contain the user's distinguished name, this is verified but not used to decide whether the user gets access to the site. Access is based solely on whether the certificate is valid and the issuer of the certificate is trusted by the server.

You can configure your server to use any one of these levels of security. Refer to your HTTP server administration manual for information on how to configure your particular software.

4.3 To make a secure connection using SSL

This section describes how you can make a secure connection using SSL.

Creating and using an SSL connection to transport data is similar to using a public key encrypted session. However, the workflow for using SSL involves additional steps to establish a secure connection:

- The user makes an HTTPS request by typing in a secure URL in the address bar of the web browser.
- The server passes its certificate to the Web Services consumer.
- The consumer checks the certificate for validity. Depending on whether the certificate has expired or the authority is trusted, the consumer may accept or reject the certificate.
- The client encrypts the certificate using the server's public key, and passes it to the Web Services provider.
- If the provider requires consumer certificates, it puts forth a request. If the consumer is unable to produce a trusted certificate, the connection is terminated.

To use HTTP over SSL (HTTPS) between a provider and a consumer, the consumer requires an X509 public key certificate. This certificate needs to be provided to the consumer object. This architecture enables the consumer to handle HTTPS requests using the public key certificate.

For more information, refer to the *SAP BusinessObjects Business Intelligence Platform Web Services Consumer SDK Developer Guide*.

4.4 Generating certificates and keystores using JDK

Certificate-based authentication provides network security by authenticating the client. Because the Web Services clients do not carry an array of information that would require the server to identify their authority, certificate-based client authentication provides sufficient security.

This section shows an example of how you can use the `keytool` utility of the JDK to generate keys, certificates, keystores, and truststores to create a self-signed certificate for the server and the client.

Example: **Script to create a self-signed certificate for the server and the client**

The following script is an example of how you can create a self-signed certificate for the server and the client.

This sample does the following:

- The server certificate is exported to the file `server.cer` and imported to the file `client.keystore` file.
- The client certificate is exported to the file `client.cer` and imported the file `server.keystore` file.
- The `server.ks` and `server.ts` files are generated.

Note:

Store these files at a convenient location such as `\https\server.ks` and `\https\server.ts` in your home directory. This practice will allow easy access while you configure web services for the Tomcat servlet container for SSL.

The script here allows the `client.keystore` file to be used by the client as the truststore to verify the certificate.

Note:

If you would like to access the server with its alias or IP address on the network, you must specify that alias while generating the key. You can have multiple certificates for the same server.

```
if not '%JAVA_HOME%' == '' goto gotJavaHome
echo You must set JAVA_HOME to point at your Java Development Kit installation
goto cleanup
:gotJavaHome
set SERVER_DN='CN=localhost, OU=X, O=Y, L=Z, S=XY, C=YZ'
set CLIENT_DN='CN=Client, OU=A, O=B, L=C, S=DD, C=EE'
rem set KSDEFAULTS=-storepass changeit -storetype JCEKS
set KSDEFAULTS=-storepass changeit
set KEYINFO=-keyalg RSA
'%java_home%\bin\keytool' -genkey -dname %SERVER_DN% %KSDEFAULTS% -keystore server.ks %KEYINFO% -keypass
changeit
'%java_home%\bin\keytool' -export -file server.cer %KSDEFAULTS% -keystore server.ks
'%java_home%\bin\keytool' -import -file server.cer %KSDEFAULTS% -keystore client.ts -alias serverkey -no
prompt
'%java_home%\bin\keytool' -genkey -dname %CLIENT_DN% %KSDEFAULTS% -keystore client.ks %KEYINFO% -keypass
changeit
'%java_home%\bin\keytool' -export -file client.cer %KSDEFAULTS% -keystore client.ks
'%java_home%\bin\keytool' -import -file client.cer %KSDEFAULTS% -keystore server.ts -alias clientkey -noprompt
:cleanup
```

```
server.cerserver.cerImportserver.cer
```

Note:

The script an example that provides only the base for a production deployment. For a production deployment, you must consider the following:

- Use of proper distinguished name or `dname` value.
- Getting certificates signed by an authorized Certificate Authority.
- Not specifying the password in the script file. If the password is not specified in the command line you will be prompted for it.
- Use of a production grade tool to manage client certificates, particularly if the server is configured to authenticate the client based on the supplied certificate.

4.5 To configure your Tomcat servlet container for SSL

1. Enable SSL for Tomcat by uncommenting the entry for the SSL connector in the `server.xml` file.

Note:

The default location for the `server.xml` file is: `%TOMCAT_HOME%\conf\server.xml`.

2. In the `server.xml` file, add the correct file path of the keystore and truststore.

The following is an example of what your `server.xml` file should look like:

```
<Connector port='8443' maxHttpHeaderSize='8192' maxThreads='150' minSpareThreads='25'
maxSpareThreads='75' enableLookups='false' disableUploadTimeout='true' acceptCount='100' scheme='https'
secure='true' clientAuth='false' sslProtocol='TLS' keystoreFile='C:\https\server.ks'
truststoreFile='C:\https\server.ts' />
```

3. Start Tomcat.

4. To verify that Tomcat has been started on your Windows machine, open the `stdout.log` file in your `webapps` directory. To verify that Tomcat has been started on your Unix, open the file `catalina.out` file the `webapps` directory . Look for the following line in your `stdout.log` or `catalina.out` file:

```
Oct 20, 2007 12:07:30 AM
org.apache.coyote.http11.Http11BaseProtocol start INFO: Starting Coyote HTTP/1.1 on
http-8443
```

This line indicates that the SSL port has been started.

5. Test the configuration.

To test the new configuration, restart the container and access the URL `https://localhost:8443` from your browser. Substitute `localhost` with your machine name and `8443` with the TCP/IP port number for Tomcat to connect to secure connections.

Alternatively, you can type the protected URL for the Portal Web Service as follows:

```
https://localhost:port/dswsbobje/services/Session?wsdl
```

Replace `localhost` with your machine name and port with the TCP/IP port number for Tomcat to connect to secure connections.

4.6 Consuming the Web Services over SSL

To consume the Web Services over SSL, you must add specific lines of code to your consumer application.

Consuming web services over SSL using Java

Add code to your web consumer application to declare a SSL URL. You must also call the `setSSL` method with the file path of the `client.ts` file as an argument to the method.

Example:

The following is an example of what your consumer application code should look like.

```
final private String urlSSL = "https://localhost:8443/dsws/services/Session";
com.businessobjects.dsws.Connection nb;
nb = new com.businessobjects.dsws.Connection(newURL(urlSSL));
com.businessobjects.dsws.SSLWrapper.setSSL("E:\\https\\client.ts");
Session m_session = new Session(nb);
String ret = m_session.getVersion();
```

Consuming web services over SSL using .NET

Add code to your web consumer application to declare a SSL URL. You must also call the `CreateFromCertFile` method with the file path of the `server.ts` file as an argument to the method. Assign the value of `X509` to the `X509Certificate` property of the `m_oCon` class.

Example:

The following is an example of what your consumer application code should look like.

```
public conststring ConnectionSessionSSL = "https://localhost:8443/dsws/services/Session";
Connection m_oCon = new Connection(ConnectionSessionSSL);
X509Certificate X509= X509Certificate.CreateFromCertFile(@"E:\https\server.cer");
m_oCon.X509Certificate = X509;
Session m_session = new Session(m_oCon);
string ret = m_session.GetVersion();
```


Scaling and performance

It is good practice to regularly assess the scalability and performance of your system and make changes to account for future growth and potential problem areas. First, you need identify the factors that can affect the current performance of your system.

This section discusses some factors that can affect the performance of your Web Services provider and consumer.

For more information about assessing your system's performance, see the *SAP BusinessObjects Business Intelligence Platform Administrator Guide*.

Network topology

The performance of Web Services is directly linked to the network topology between the Web Service server and its consumer. Because all information is transmitted through the network, the size of the information being transmitted has the same effect on performance as the power of the machines involved. If a bottleneck occurs, it is likely to do so at the network level.

Therefore, when planning your Web Services deployment, you should consider the best ways to make your information available to your consumers. For example, it is more efficient to send small reports in HTML than in PDF format.

Server performance

The performance of the BI platform servers will affect the performance of the Web Services. If the performance of the servers are optimized this will improve the performance of the Web Services.

Network round-trip

The methods in the consumer classes retrieve and send information using network calls to the Web Services providers installed in a BI platform deployment. Therefore, using these methods will impact network traffic. You should plan to use these methods as efficiently as possible in order to reduce the number of network calls to the Web Services.

Since our API has a coarse-grained design, you can improve your application performance by doing only one network round-trip per workflow. It is also recommended that you request exactly the information you need and nothing more. For example, if you want to display the first level of the BICatalog, you should just request the first level and not all levels. This is important since this will minimize network traffic.

You can reduce the calls from the consumer application to the Web Service by caching in the user's session information, the document lists, indexes, and document information (such as drill information and prompts) as much as possible.

Scalability

To improve scalability in your deployment scenario, you should look into adding load-balancing between your application servers. You can, for example, balance loads over servers by strategically distributing high-traffic system modules (or processes) across several machines, or in some cases, by directing a greater proportion of transactions to higher-capacity servers. You can dedicate certain servers to a single type of process, or provide for several types of processing on several servers.

More Information

Information Resource	Location
SAP product information	http://www.sap.com
SAP Help Portal	<p>http://help.sap.com/businessobjects</p> <p>Access the most up-to-date English documentation covering all SAP BusinessObjects products at the SAP Help Portal:</p> <ul style="list-style-type: none"> • http://help.sap.com/bobi (Business Intelligence) • http://help.sap.com/boepm (Enterprise Performance Management) • http://help.sap.com/boeim (Enterprise Information Management) <p>Certain guides linked to from the SAP Help Portal are stored on the SAP Service Marketplace. Customers with a maintenance agreement have an authorized user ID to access this site. To obtain an ID, contact your customer support representative.</p> <p>To find a comprehensive list of product documentation in all supported languages, visit: http://help.sap.com/boall.</p>
SAP Support Portal	<p>http://service.sap.com/bosap-support</p> <p>The SAP Support Portal contains information about Customer Support programs and services. It also has links to a wide range of technical information and downloads. Customers with a maintenance agreement have an authorized user ID to access this site. To obtain an ID, contact your customer support representative.</p>
Developer resources	<p>http://www.sdn.sap.com/irj/sdn/bi-sdk-dev</p> <p>https://www.sdn.sap.com/irj/sdn/businessobjects-sdklibrary</p>
SAP BusinessObjects articles on the SAP Community Network	<p>http://www.sdn.sap.com/irj/boc/articles</p> <p>These articles were formerly known as technical papers.</p>

Information Resource	Location
Notes	https://service.sap.com/notes These notes were formerly known as Knowledge Base articles.
Forums on the SAP Community Network	https://www.sdn.sap.com/irj/scn/forums
Training	http://www.sap.com/services/education From traditional classroom learning to targeted e-learning seminars, we can offer a training package to suit your learning needs and preferred learning style.
Consulting	http://www.sap.com/services/bysubject/businessobjectsconsulting Consultants can accompany you from the initial analysis stage to the delivery of your deployment project. Expertise is available in topics such as relational and multidimensional databases, connectivity, database design tools, and customized embedding technology.

Index

C

- clustering 16
- configuring
 - dsws.properties 12
 - services 12

D

- deployment 11
 - clustering 16
 - configuration properties 12
 - configuring services 12
 - verification 11

- deployment (*continued*)
 - WSDL 11
- document history 5
- dsws.properties file 12

E

- enabling
 - services 12

M

- migration 9

P

- performance 27

S

- SSL 19

W

- Web Farms 16

