Mindtree
A Larsen & Toubro Group Company

SOLUTION DESIGN

ENABLING CORBA SSL FOR SAP BO 4.2

Welcome to possible

A Mindtree White Paper | July 2021

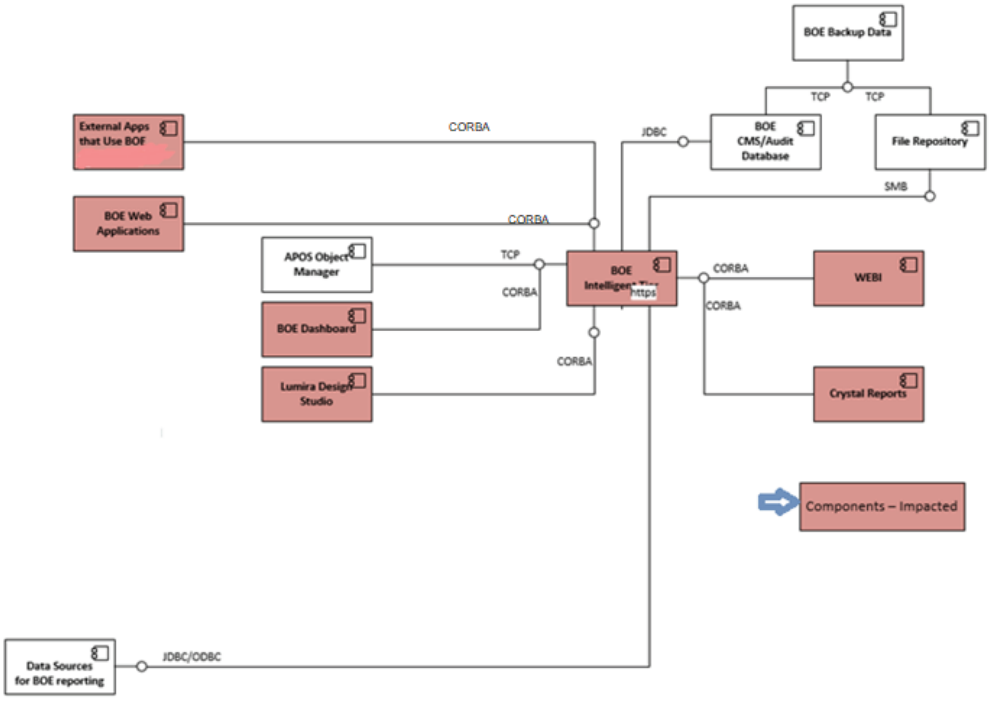
Table of contents

ABSTRACT

SAP Business objects is one of the leading reporting platforms in the market, which will give you extensive reporting solutions for your business needs.

The intention of this document is to provide an overview of the security concerns with the SAP Business objects reporting platform and tools while communicating internally. This document also explains how to overcome this problem by implementing CORBA SSL (procedural steps suggested by SAP to secure internal communications within the SAP Business objects platform)

COMPONENT DIAGRAM



The above diagram shows sample business objects architecture and the components:

- The above diagram shows the connectivity protocols of different components to the BOE Intelligent tier in Business Objects architecture
- Components like APOS Object manager (External component used for Business objects administration activities) are using TCP protocol
- Business objects back up data will use TCP/IP connectivity.
- BOE central management console (Used for BO administrator activities) and Audit database (Database where BO platform related data will store) uses JDBC connection
- File repository will use the SMS protocol to connect with the business intelligent tier
- External APPS that use BO, BO WEB applications, BO Dashboards (Visualization tool in Sap BO), Lumira design studio (Visualization tool in SAP BO), WEBI, and CRYSTAL (Reporting tools in SAP BO) will use the CORBA SSL

WHAT IS CORBA SSL:

CORBA stands for Common Object Request Broker Architecture, it is the standard for varied computing. SSL stands for Secured Socket Layer protocol, CORBA SSL is used to encrypt the server communications. SAP Business Objects 4.2 should be encrypted to avoid any security issues. We need to enable CORBA SSL in Business Objects. This would affect all the servers, clients, and 3rd party applications communicating with SAP Business Objects.

ENABLING CORBA SSL

CONFIGURING SERVERS WITH SSL

The SSL protocol is used with the BIBO platform deployments for all the network communications between clients and servers.

To implement the SSL protocol for all server communications, the following steps need to be implemented:

- Setup the SSL enabled for BI platform
- Produce files (key and certificate) for each environment (Dev, QA, PROD) in deployment
- Setup the location path of these files in Central Configuration Manager (CCM) and web application server (Tomcat), or configure the SSL with certificates that are managed by a certificate authority within the organization.

Note:

We need to configure the SSL for the reporting tools in business objects before connecting the SSL enable Business objects platform, else we will get an error.

CONFIGURING SERVERS WITH SSL

When a certificate or a certificate generation request needs to be created, one will need to create default files for configuration to avoid repetitive addition values.

Note:

Rules to be followed while creating the default configuration file:

- Provide the left hand side values, which are showed exactly as below
- Values that are mentioned on the left hand side need to be case sensitive
- Only one space is allowed between the value and an 'equal to' (=) sign
- Ensure there are spaces on right hand side after the values

Please follow the below steps to create the configuration file with default name N**ame.cnf**

1. Open the new text document in a text editor
2. Add values as mentioned in the below image:

```
CA_Common_Name = rootnm
CA_Country = DE
CA_State = BW
CA_Locality = RRR
CA_Email = root@gmail.com
CA_Unit = root_u
CA_Expiration[YYMMDD] = yymmdd
User_Expiration[YYMMDD] = yymmdd
User_Country = IN
User_State = KA
User_Locality = BLR
User_Organization = SSS
User_Unit = Unit
User_Common_Name = UserName
```

3. Save the file with the name Name.cnf at <INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\win64_x64 in Windows

GENERATING CERTIFICATE AND KEY FILES

To set up the SSL protocol to communicate with the server, the command line tool called 'GENPSE' should be used to create certificates and key files for each environment.

Note:

- To configure the SSL in machines within the deployment and machines that have a thick client, one needs to recreate the certificates and then run the SSLCONFIG command line tool
- To ensure maximum security, we need to protect all private keys. These keys are not supposed to be sent through unsecured communication channels
- The certificates created for the lower versions are not supported in Bi 4.2 SP4. The new certificates SP4 should be created for BI 4.2 SP4, because the minimum key strength for the certificates have been increased to 2048.
- Generating files in one server within the cluster is enough for sending to the certification authority. We can use the same files and certificates in all servers within the environment

CONFIGURING SSL WHERE CERTIFICATES ARE MANAGED BY CERTIFICATION AUTHORITY

In the organization where they have separate certification authority, who will manage and issue the security certificate we need to follow the below steps for generating the necessary docs to share with certification authority and get the certificates to enable SSL:

- Generate the signing request using the GENPSE command to get the signed certificates from a third-party authority.
- Use GENPSE tool to execute commands to generate certificate signing requests, and please provide the necessary information when prompted.

Follow the below steps to generate a certificate signing request:

Note:

Please create a default configuration file with ".cnf" extension (Example: XYZ.cnf), and provide the default values for the information while generating certificates. This will provide relief from the addition of information details for each certificate every time. The process of creating default configuration files is mentioned in the "Generate default configuration files section" in the document.

- Open the path <INSTALLDIR>\ SAP BusinessObjects Enterprise XI 4.0\win64_x64 in Windows and<INSTALLDIR>/sap_bobj/enterprise_xi40/linux_x64 in UNIX.
- Run the command:
    - In Windows: GenPSE.exe gencsr <csrname.p10> <Name.key> <private key password.txt> <path to Name.cnf>
    - In Unix: GenPSE.sh gencsr <csrname.p10> <Name.key> <private key password.txt> <path to Name.cnf>

Command	Function
GenPSE.exe or GenPSE.sh	Start the cryptography tool
gencsr	To generate Certificate Signing Request
<csrname.p10>	Certificate Signing Request filename
<Name.key>	Server private key filename
<private key password.txt>	Passphrase for server private key file
<path to Name.cnf>	Path of the default configuration file

3. Enter the following information:

- Enter private key passphrase to set
- Re-enter private key passphrase to confirm
- Country Name
- State or Province Name
- Locality Name
- Organizational Unit Name
- Common Name
- Email Address

4. CSR file in p10 format, server private key, and the passphrase file are generated and stored at <INSTALL-DIR>\SAP Business Objects Enterprise XI 4.0\win64_x64 in Windows. For generating the signed certificate, we need to send the generated CSR file to certification authority.

PSE FILE GENERATION

PSE files need to be created to get the certificates from the certification authority of the organization when certification authority manages the latter. To generate a PSE file, please follow the below steps:

- Navigate <INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\win64_x64.
- Open Command line prompt and run set "SECDIR=" in windows and for Linus export "SECUDIR="
- Run sapgenpse import_p8 -p <file_path_PSE> -c <file_path_server_certificate> -r <file_path_CA_certificate> -z <file_path_passphrase_text_file> <file_path_server_key>.

Please use the below table to understand the commands using GENPSE:

Command	Description
sapgenpse	Start the cryptography tool
import_p8	Create a new PSE file from a PKCS#8 format private key (optionally protected by PKCS#5 password-based encryption) along with all necessary X.509 certificates
-p <file_path_PSE>	File path to the new PSE file that is created
-c <file_path_server_certificate>	File path to the server certificate
-r <file_path_CA_certificate>	File path to the CA certificate
-z <file_path_passphrase_text_file>	File path to the passphrase text file
<file_path_server_key>	File path to the server private key file

Example

sapgenpse import_p8 -p C:\SSL\cert.pse -c C:\SSL\servercert.der -r C:\SSL\cacert.der -z C:\SSL\passphrase.txt C:\SSL\server.key

- Do not enter any password. Press enter for an empty password when it prompt for password.
- Add user credentials for the created pse file.

CONFIGURING SSL FOR THE BI SERVER

To configure the Business objects servers for CORBA SSL please follow the below steps

1. Setup the server:

- In Windows, open the command prompt (run as Administrator). In Linux/UNIX, open a terminal session as a BI user
- Create the following folder structure

  Windows MKDIR <Path folder>:\SSL

  Linux

  <PATHTOFOLDER>/ssl

- Open the folder in business objects installable, which contains GENPSE from the command prompt:

CD "<Install Directory>\SAP BusinessObjects\SAP BusinessObjects

cd < Enterprise XI 4.0\win64_x64"

- Now create text file with the name SSL [Change the extension to .cnf]
- Open the SSL.cnf file in edit mode and provide default values as shown below. Then, save the file.

CA_Common_Name = SAP Company

CA_Country = US

CA_State = GA

CA_Locality = APH

CA_Email = xyz@sap.com

CA_Unit = Product Support

CA_Expiration [YYMMDD] = 201231

User_Expiration[YYMMDD] = 201231

User_Country = US

User_State = GA

User_Locality = APH

User_Organization = DBS

User_Unit = PS

User_Common_Name = UserName

2. Creating the required certificate and key files for BI server:

The steps to create the required certificate are simplified in 4.2 Service Pack5, when compared to older versions. Now, it is a single step process to generate self-signed certificates. By using the SAP GENPSE tool, we can generate all certificates required for SSL. GENPSE is the command line tool to execute the SSL commands abundant related to the public key structure. To generate X.509 certificates, the GENPSE tool needs to be used. The tool is also used for creating certificate signing requests and the PSE file, which will be used in the CORBA SSL setup. This procedure is based on CommonCryptoLib, which supports SHA-2 hashing provided by SAP's cryptographic library, which engenders all self-signed certificates and keys using SAP GENPSE in the windows command prompt (open command prompt as administrator):

GenPSE selfsigned cert.pse servercert.der cacert.der server.key passphrase.txt ssl.cnf

Note: When the data is created, keep in mind that the ROOT CA Certificate Communal Name (CN) and Server PSE Common name must be different. The naming for ROOT CA certificates is in line with the host name of the server.

- Provide the required information when prompted
- Certificate names should start with S
- Date formats should be in YYMMDD (we can give up to year 2049)
- Provide an empty password when prompted
- Make sure the CA certificate and PSE files have different names

The PSE file and the certificate are created and placed in 64 bit folders. The below files are generated using the following steps:

Cacert.der ->CA Certificate file trusted

Servercert.der-> Server file certificate

Server.key-> Private key server file

Cert.pse-> Pse file server

passphrase.txt-> Server private key for decrypting passphrase

The PSE file and the certificate are created and placed in 64 bit folders. The below files are generated using the following steps:

Cacert.der ->CA Certificate file trusted

Servercert.der-> Server file certificate

Server.key-> Private key server file

Cert.pse-> Pse file server

passphrase.txt-> Server private key for decrypting passphrase

Once we generate all the files, please copy them to the SSL folder created in step 1 (These can be copied manually or the command prompt can be used)

COPY cacert.der C:\ssl COPY servercert.der C:\ssl COPY server.key C:\ssl COPY cert.pse C:\ssl

COPY passphrase.txt C:\ssl

3. **Configuring SSL Protocol for the BI Server:**

- Open CCM, stop the server intelligence agent, and open properties
- Navigate to the protocol tab in properties tab
- Check the enable SSL Business Objects
- Provide the certificate files directory path where the certificates need to be stored (example: C:\SSL\cacert.der)

SSL Trusted Certificate-(C:\SSL\cacert.der)->SSL trusted certificate

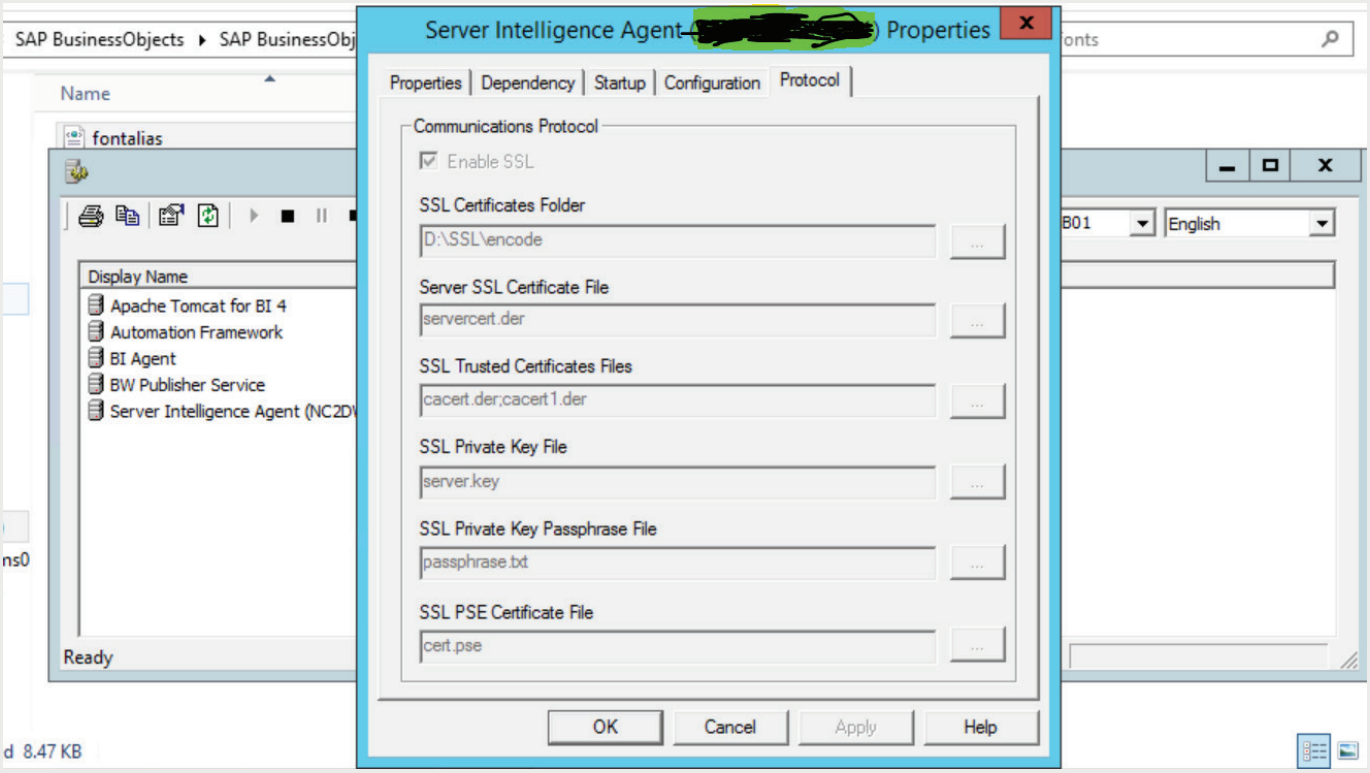
SSL Private key file- (C:\SSL\server.key)->SSL private key file used to access the certificate

SSL passphrase file- (C:\SSL\passphrase.txt)->Used to access the private key

SSL PSE certificate- (C:\SSL\cert.pse)->Which will contain information about the trusted and server certificate

- Start the SIA

  Note: Even if the server is configured with SSL, one may get an error when trying to login to CMS using the CCM tool. Please refer the Appendix section for some reference default errors



SETTING SSL FOR THICK CLIENT AND .NET OR SDK APPLICATIONS FOR SERVER SIDE SSL COMMUNICATION

For configuring the thick client for SSL, use the command sslconfig.exe(On Linux use boe_sslconfig)

- On the machine where the thick client is:

  For Servers which use 64-bit clients, use the win64_x64(In Linus linux_x64) directory : CD "<Install Directory>\SAP Business Objects\SAP Business Objects Enterprise XI 4.0\win64_x64

Note: Command Prompt must be opened in 'Run as Administrator' mode

  CD ""<Install Directory>\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86"

- Run the following command:

  sslconfig.exe -dir C:\SSL -mycert servercert.der -rootcert cacert.der -mykey server.key -passphrase passphrase.txt -psecert cert.pse -protocol ssl

- Once the command is run successfully, there is a confirmation that the protocol has been set to SSL.

Note: The registry needs to be configured if one were to use Linux\UNIX for thick clients (ccm.sh)

CONFIGURING WEB APP SERVER (TOMCAT) TO COMMUNICATE WITH BI SERVER WHICH ENABLED SSL

To set up the Web application Servers (Apache Tomcat) for communicating with the BI servers that are SSL configured, we need to add the below properties to the Java options of Tomcat:

Example:

-Dbusinessobjects.orb.oci.protocol=ssl -DcertDir=c:/ssl - DtrustedCert=cacert.der -DsslCert=servercert.der -DsslKey=server.key - Dpassphrase=passphrase.txt

To configure an Apache Tomcat server which is running on the Windows-based BI servers:

- Click on "Windows-> Search for Tomcat-> Open Tomcat->Tomcat configuration-> Java" add the below entries at the end of "Java options"

- Enter the below values to the Java options text box (make sure that there are no trailing or preceding spaces)

Dbusinessobjects.orb.oci.protocol=ssl
DcertDir=C:/SSL
DtrustedCert=cacert.der
DsslCert=servercert.der
DsslKey=server.key
Dpassphrase=passphrase.txt
Dpsecert=cert.pse

3. After copying the above properties to the JAVA options, click on OK to save. Once this is done, Tomcat needs to be restarted for the changes to work. For a non-windows environment, the same steps (like above steps i, ii, iii) need to be followed.

Example (<sap_bobj location>/tomcat/bin/setenv.sh):

JAVA_OPTS="$JAVA_OPTS -Dbusinessobjects.orb.oci.protocol=ssl - DcertDir=<folderpath>/ssl -DtrustedCert=cacert.der -DsslCert=servercert.der - DsslKey=server.key -Dpassphrase=passphrase.txt -Dpsecert=cert.pse" export JAVA_OPTS

THICK CLIENT CONFIGURATION CHANGES

To configure SSL for thick clients in the Business Intelligence Platform, navigate to <INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\win64_x64 & <INSTALLDIR>\SAP BusinessObjects Enterprise XI 4.0\win32_x86 and run the following command by replacing the variables with the specific values:

"sslconfig.exe -dir C:\SSL -mycert servercert.der -rootcert cacert.der -mykey server.key -passphrase passphrase.txt -psecert cert.pse -protocol mixed"

<certdir> is the directory where the following SSL files are stored. <sdkcert> is the certificate (DER format). <rootcert>is the root certificate (DER format).<privatekey> is the key file. <passphrase> which is a passphrase plain text used to decrypt the generated private key. <psecert> the pse file which contains the trusted certificate and server certificate information.

Note: After running the command, the SIA needs to be rebooted for a successfull connection to the server from UDT.

Note: After running the command, the SIA needs to be rebooted for a successfull connection to the server from UDT.

INFORMATION DESIGN TOOL – CONFIG CHANGES

File path: <INSTALLDIR>\SAPBusinessObjects Enterprise XI 4.0\win32_x86

Edit the file mentioned as "InformationDesignTool" (file type configuration) and add the following switches:

-Dbusinessobjects.orb.oci.protocol=ssl

-DcertDir=D:\SSLCert

-DtrustedCert=cacert.der

-DsslCert=servercert.der

-DsslKey=server.key

-Dpassphrase=passphrase.txt

Please make sure to replace the values with the appropriate ones after the installation.

-DcertDir Directory to hold the SSL Certificates and Keys

-DtrustedCert The CA Certificate (shared for all certificates), additional CA Certificates can be added with semi-colon as separator (;) Eg. -DtrustedCert=cacert1.cer;cacert2.cer;cacert3.cer

-DsslCert The SSL Certificate (usually a client certificate created and signed with the CA Certificate)

-DsslKey The SSL Key (client key)

-Dpassphrase The file containing the passphrase for the SSL Key

DISABLING CORBA SSL

Corba SSL - disable it on the cluster, web app server, and every client machine

Windows: Repeat on each node in the cluster.

- Open the CCM and stop the SIA
- Choose the Protocol tab on the SIA properties.
- Uncheck "Enable SSL"
- Open a command prompt
- Change to the install folder\SAP BusinessObjects Enterprise XI 4.0\win32_x86
- Run: sslconfig.exe -protocol default
- Now cd ..\win64_x64
- Run: sslconfig.exe -protocol default
- Open the Tomcat configuration manager (Start -> Tomcat -> Tomcat Configuration)
- Stop Tomcat
- Go to the Java tab
- In the large box, remove the java options for CORBA SSL, which will look something like this: -Dbusinessobjects.orb.oci.protocol=ssl -DcertDir=<SSLFILEPATH> -DtrustedCert=cacert.der -DsslCert=servercert.der -DsslKey=server.key -Dpassphrase=passphrase.txt
- Click OK to close this window.
- Start Tomcat

DEPLOYMENT INFORMATION

INFRASTRUCTURE IMPACT

New: BOE Servers

New: Tomcat Servers

New: BO Client Tools installable (INI files & Batch Scripts for SSL)

DEPLOYABLE FILES

CONFIGURATION

INI file changes (Client tools)

Enable SSL via Batch scripts

STATIC CONTENT

N/A

RISKS

ASSUMPTIONS

- No performance Issues would be encountered by enabling the CORBA SSL
- The application packaging team can develop the scripts to update the client tools and run the batch script in all laptops that had BO client tools (Silent batch)

OPERATIONAL RISKS

For configuring the thick client for SSL, use the command sslconfig.exe(On Linux use boe_sslconfig)

- CORBA SSL:
  - BO Admin team needs to work with SAP and work on the POC to work with CA certificates
  - INI files changes and batch script need to be run for SSL enabling in client tools
- Whenever there is a need to upgrade patches or service packs or versions, it is necessary to disable the CORBA SSL, apply the patches and enable the SSL again.
- Once you enable the CORBA SSL and restart the SIA, make sure all the services are up and running in the CCM and then try logging to the CMC or BI LAUNCHPAD.

CONCLUSION

In an abstract, I have described about the problem statement of SAP Business objects platform security concerns for internal communications with the reporting tools of BO. I have explained the solution provided by the SAP to secure the internal communications of SAP BO using CORBA SSL. I have also explained the CORBA SSL implementation steps in details with all necessary information in this document. By referencing this document, you can set up the CORBA SSL in your SAP Business Objects environment to secure BO reporting tools communications within the Business objects environment.

APPENDIX A: SECURITY CONSIDERATIONS

Authentication

- Inactivity Timeout: Stateless

Authorization
URL-based security constraints defined in web.xml

Sensitive Data Handling
N/A

Input Validation/Output Encoding
The application can be accessed directly from the browser.

File Upload
N/A

Web Services (published)

- Web Security for RESTful Web Services (Document not yet available)
  - Services are protected by basic authentication
  - Will have its own AD Group and credentials in PREPROD and PROD

Logging

Logging information captured by the application includes (at a minimum) the following:

- Date Time, Log Level, UserID, Thread Id, Class Name, Log Message

Pramod Kumar Matam

Module Lead

Pramod is an SAP reporting Consultant, and has vast experience in SAP Business objects platform reporting and administration.