

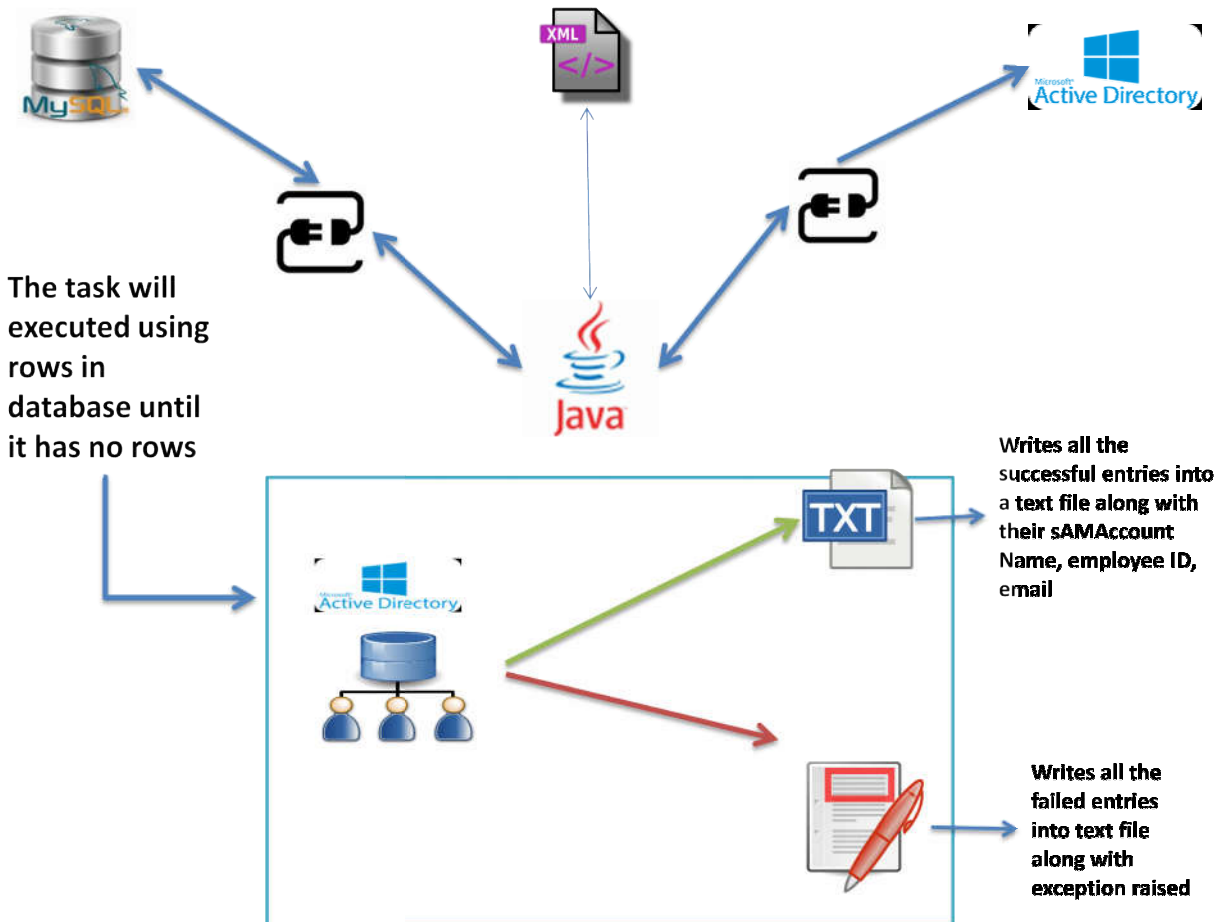
Inserting users from SQL data to Active Directory

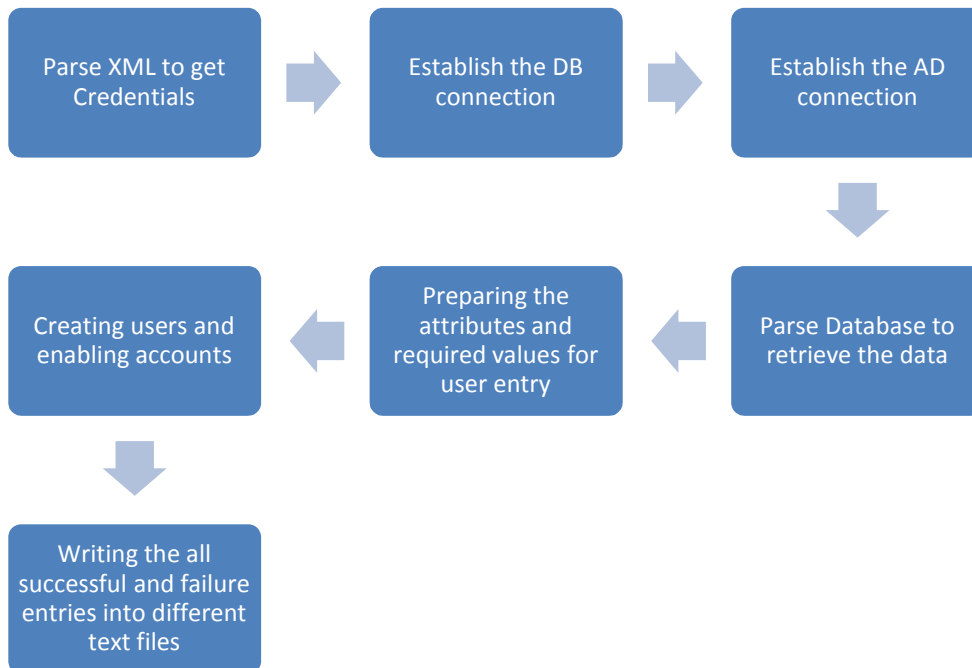
Contents

1. Requirement	2
2. Prerequisites:	3
3. Environment used	3
4. Steps following	3
5. Parsing the XML file	5
6. Establishing the database connection	6
7. Establishing the directory connection	7
8. Parsing the SQL	8
9. Making the entries	8
10. Adding users to Active directory	9
1. addUsers()	10
2. ModifyPassword()	10
3. writeData()	10
4. main()	11
11. Table with package data	11

1. Requirement

This document explains about the program which is used to adding users to active directory from the sql data.





2. Prerequisites:

The document explains about creating users in active directory from SQL data using the java program.

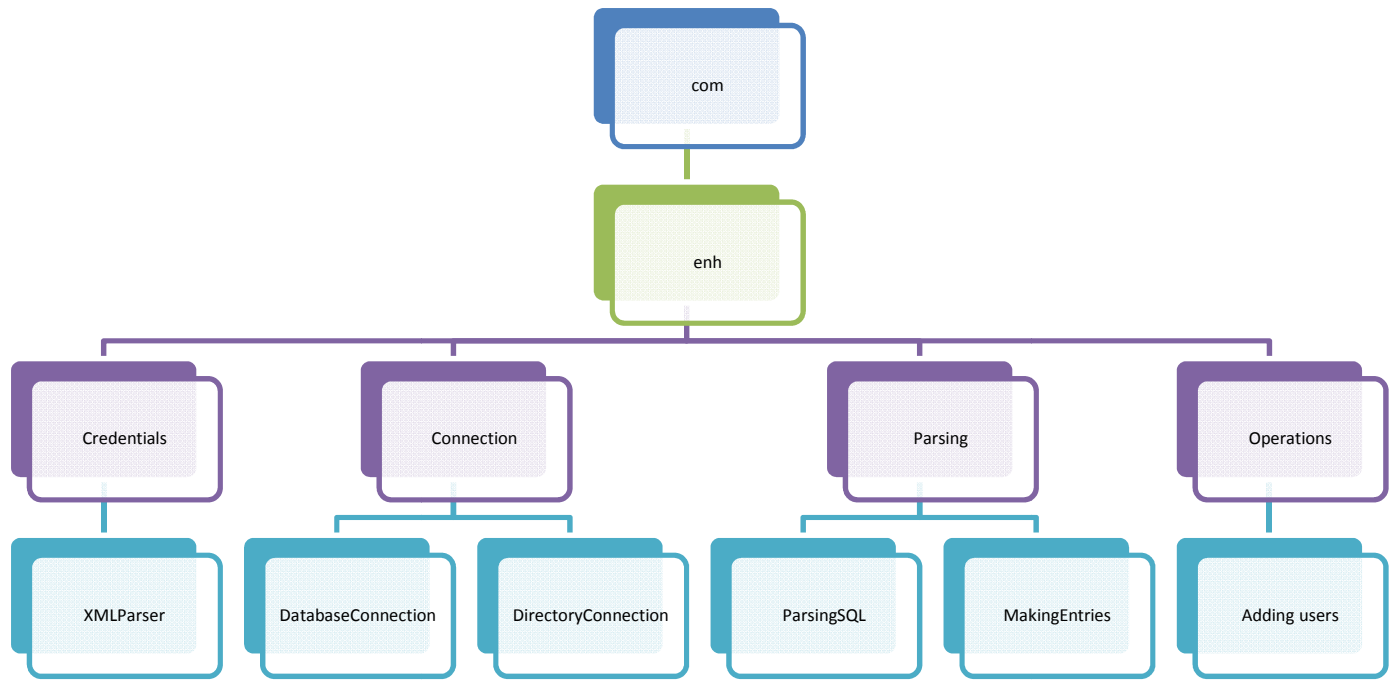
Before going through this document better to understand about the Active directory, JNDI.

3. Environment used

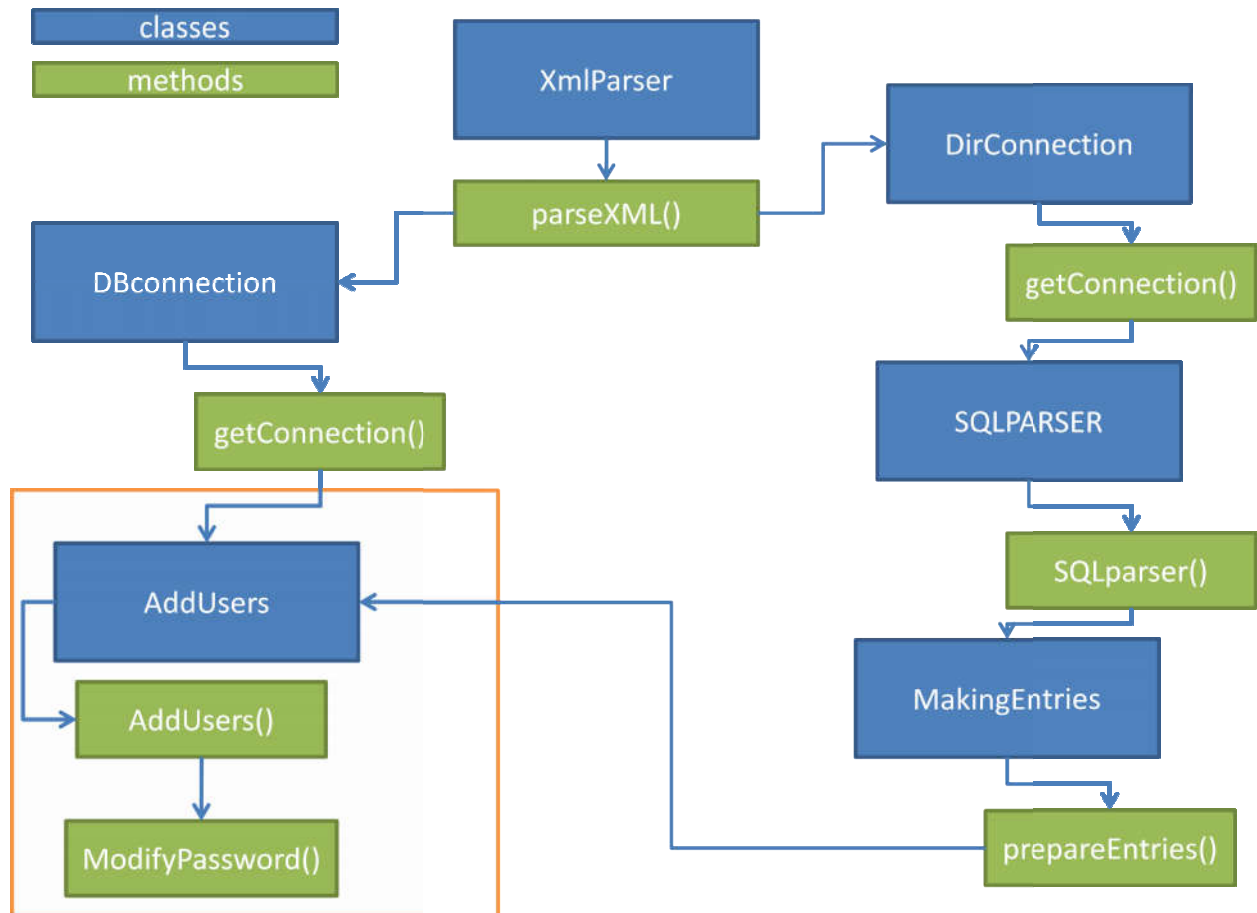
- database - mysql
- programming language - java SE 8
- scripting - power shell
- operating systems used
 - Windows server configured with the Active directory and Exchange server
 - Oracle Linux

4. Steps following

The following diagram explains about the package hierarchy of the classes we are creating.



The following diagram explains about the classes and its functionality in the program and the flow we are following in this program.



From the above diagrams we can say that the following are the mandatory steps in this program

- Parsing the xml file
- Establishing the database connection
- Establishing the active directory connection
- Parsing the SQL data
- Preparing the entries for adding users
- Adding users with password

5. Parsing the XML file

In this scenario XML file is containing the credentials to connect to both database and active directory.

We are creating the ParsingXML in the package com.enh.connections

To parse the XML file we need the following packages:

```
import java.io.File;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import org.xml.sax.SAXException;
import javax.xml.parsers.ParserConfigurationException;
import java.io.IOException;
import java.util.HashMap;
```

Using the File class we can mention the file which is having the credentials information.

In this class we are creating a method parseXML which will perform the parsing and store the information in a HashMap for individual like for database which is dbCredentials and for active directory mapped as dirCredentials.

The class is developed such as way that the xml file should be located at the directory where all the java files are created.

Reading the data according to the nodes and elements presented in the XML file.

6. Establishing the database connection

The database connection is established using the class DatabaseConnection and its method getConnection().

This class is created in the package com.enh.connections.

To establish the connection we are using the credentials obtained by the XMLParser class.

To establish connection requires some interfaces and its method which presents in the jdbc connector.

To establish the connection we need to import the following classes.

```
import com.enh.credentials.XmlParser;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Connection;
import org.xml.sax.SAXException;
import javax.xml.parsers.ParserConfigurationException;
import java.io.IOException;
```

getConnection requires the credentials for connection. For that we need to call the parseXML method using the XMLparser object.

This method enables us to create connection using DriverManager class.

7. Establishing the directory connection

The DirectoryConnection class presented in the package com.enh.connection explains about the connection between the active directory and java program.

In this scenario we are accessing the active directory in SSL mode. To enable the SSL authentication for active directory refer the following [link](#).

The following packages are to be imported to establish the connection

```
import com.enh.credentials.XmlParser;  
import java.util.Hashtable;  
import javax.naming.Context;  
import javax.naming.NamingException;  
import javax.naming.directory.InitialDirContext;  
import javax.naming.directory.DirContext;  
import org.xml.sax.SAXException;  
import javax.xml.parsers.ParserConfigurationException;  
import java.io.IOException;
```

getConnection() Method presented in the class DirectoryConnection enables us to connect the active directory for that we need to get the credentials required.

To get the credentials we need to call the parseXML () method using the XMLparser object.

Using the HashMap which contains directory server data we can connect to active directory

To establish the connection we need to build an environment which can store the details and credentials required.

We are using the Hashtable as environment as the environment.

After adding the necessary details to environment we need to create the context required for the active directory.

It can be done using DirContext object.

8. Parsing the SQL

The class ParseSQL presented in package com.enh.parsing explains about parsing the SQL and storing into ResultSet.

The following packages are to be imported to create this class

```
import com.enh.connection.DatabaseConnection;
import com.enh.credentials.XmlParser;
import java.sql.*;
import org.xml.sax.SAXException;
import javax.xml.parsers.ParserConfigurationException;
import java.io.IOException;
```

For parsing the SQL data we need to use the connection object created in the DatabaseConnection class.

We are accessing connection object using the DatabaseConnection object.

Before creating the resultset we need to create the statement object that helps to pass the queries to the resultset.

In this case we are using statement as scrollable and scroll sensitive because we are using the result set in multiple ways.

After creating the statement object we need to create the result by passing a sql query required for it.

In this case we are retrieving the all data from the table so that we need to pass the query as below format "**select * from *table_name***"

we need to pass the above query for the resultset using the statement object which we created.

9. Making the entries

The class MakingEntries is presented in the package com.enh.parsing.

The class contains the method prepareEntries() which prepares the entries in active directory format.

The following packages are to be imported to write this class

```
import com.enh.credentials.XmlParser;
import com.enh.parsingSQL.ParseSQL;
import javax.naming.*;
import javax.naming.directory.*;
import java.util.HashMap;
```



```
import java.sql.SQLException;
import org.xml.sax.SAXException;
import javax.xml.parsers.ParserConfigurationException;
import java.io.IOException;
import javax.naming.Context;
import javax.naming.NamingException;
import javax.naming.directory.InitialDirContext;
import javax.naming.directory.DirContext;
import javax.naming.directory.Attributes;
```

After importing the packages we need to create the object for ParsingSQL class which is used to access the resultset. Before adding the data to active directory desirable format it's better to store the resultset data into a hashmap.

After storing the data into hashmap we need to create an object for Attribute which can map desirable attribute to resultset data.

Store the relevant data for the user such as sAMAccountName, email, first name, LastName, user principal name etc

After adding the all details to the Attribute object we need to create the Attributes object using the Attribute object.

10. Adding users to Active directory

The class AddUsers is presents in the package com.enh.operations and having following methods.

- addUsers()
- changePassword()
- writeData()
- main()

The following are the packages to be imported to write this class.

```
import com.enh.parsingSQL.MakingEntries;
import com.enh.connection.DirectoryConnection;
import com.enh.connection.DatabaseConnection;
import com.enh.parsingSQL.ParseSQL;
import com.enh.credentials.XmlParser;
import java.sql.SQLException;
```

```
import org.xml.sax.SAXException;
import javax.xml.parsers.ParserConfigurationException;
import java.io.IOException;
import java.io.FileOutputStream;
import java.io.PrintStream;
import java.io.FileNotFoundException;
import java.io.UnsupportedEncodingException;
import javax.naming.*;
import javax.naming.NamingException;
import javax.naming.directory.*;
```

after importing the packages we need to create the objects for every class we are using in this package so that we can access every method in this package in a more proper way.

1. addUsers()

This method is used to add the users to active directory in hierarchical manner.

In this method we are accessing the result set so that we need to call parseSQL method using the ParseSQL object.

Then we need to call prepareEntries method using the MakingEntries object.

Now time to add user to the active directory, To add the users to active directory we need to access the DirContext which we have created in the, DirectoryConnection class hence we need to access that DirContext using the object created for DirectoryConnection.

A user has been created with no password.

Then it moves to modifying the password , after that we need to write the details of the users which we have created and error for adding users into a text file.

2. ModifyPassword()

this method updates the password with some given string using some interface objects required for active directory.

using the directoryConnection and we can change password using the method modifyAttributes.

3. writeData()

In this method we are writing data of the user which successfully inserted into active directory in the following format.

employee_number , sAMAccountNumber, email id

4. main()

This is method where every iteration starts, calling the method addUsers from this method start the program and users will be added to the active directory.

11. Table with package data

The following table explains about the classes, methods, variables, exceptions handled in this document

	package	Class name	Variables	Exceptions	methods
1	com.enh.credentials	XMLParser	DbCredentials; DirCredentials;	ParserConfigurationException, SAXException, IOException	void parseXML()
2	com.enh.connection	DatabaseConnection	Connection dbconnection;	ClassNotFoundException, SQLException, ParserConfigurationException, SAXException, IOException	void getConnection()
3	com.enh.connections	DirectoryConnection	DirContext ctx; Hashtable environment;	NamingException, ParserConfigurationException, SAXException, IOException	void getConnection()
4	com.enh.parsingSQL	ParseSQL	public static ResultSet resultset ; public Statement statement ;	SQLException, ClassNotFoundException, ClassNotFoundException, SQLException, ParserConfigurationException, SAXException, IOException	void parseSQL()
5	com.enh.parsingSQL	MakingEntities	public String	ClassNotFoundException, SQLException,	void prepare

			dn; public Attribute s entry; public int orgCount;	ParserConfigurationException, SAXException, IOException	Entries ()
6	com.enh.o perations	Addusers	Directory Connectio n directory PrintStre am ADuserWri ter PrintStre am exception Writer; String completeU serName	NamingException, SQLException, ClassNotFoundException,Un supportedEncodingException, FileNotFoundException, ParserConfigurationException, SAXException, IOException	void addUser s ()
7	com.enh.o perations	Addusers		NamingException, UnsupportedEncodingException, ParserConfigurationException, SAXException,IOException	ModifyP assword ()
8	com.enh.o perations	Addusers		FileNotFoundException	writeDa ta ()
9	com.enh.o perations	Addusers		NamingException, SQLException, ClassNotFoundException,Un supportedEncodingException, FileNotFoundException, ParserConfigurationException, SAXException, IOException	public static void main ()

