

ETL Validator Setup Guide

Table of contents

Introduction	3
License	3
ETL Validator Architecture	4
Client-Server Architecture	4
Topology	6
ETL Validator Setup	7
System requirements	8
Database Requirements	8
Server Setup	9
Client and Repository/Workschemata Setup	11
Connection Setup	14
Certification Matrix	15
Replicating ETL Validator Setup	16
Copy Repository and Workschemata	16
Setup ETL Validator Server	16
Configuring Client to new Repository	17

Introduction

ETL Validator helps automate the testing of data integration and data warehouse projects. It makes use of our patented ELV Architecture (Extract, Load and Validate) to extract test data from heterogeneous data sources, load it to a test data store and run validations on the test data set.

Datagaps also offers a comprehensive regression and performance testing tool called BI Validator for testing BI applications built using OBIEE.

This help documentation is designed so you can quickly install and setup ETL Validator.

License

Copyright © 2010, 2015, datagaps inc. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable datagaps license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987).

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

ETL Validator is a registered trademarks of datagaps inc and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Datagaps is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Datagaps is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Datagaps is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Copyright © 2010, 2015, datagaps inc. All rights reserved.

ETL Validator Architecture

ETL Validator architecture consists of a windows based client, a J2EE server (tomcat) and a repository (Oracle or Postgresql database). It makes use of our patented ELV Architecture to extract and validate test data from multiple sources. The links below provide details on ETL Validator architecture.

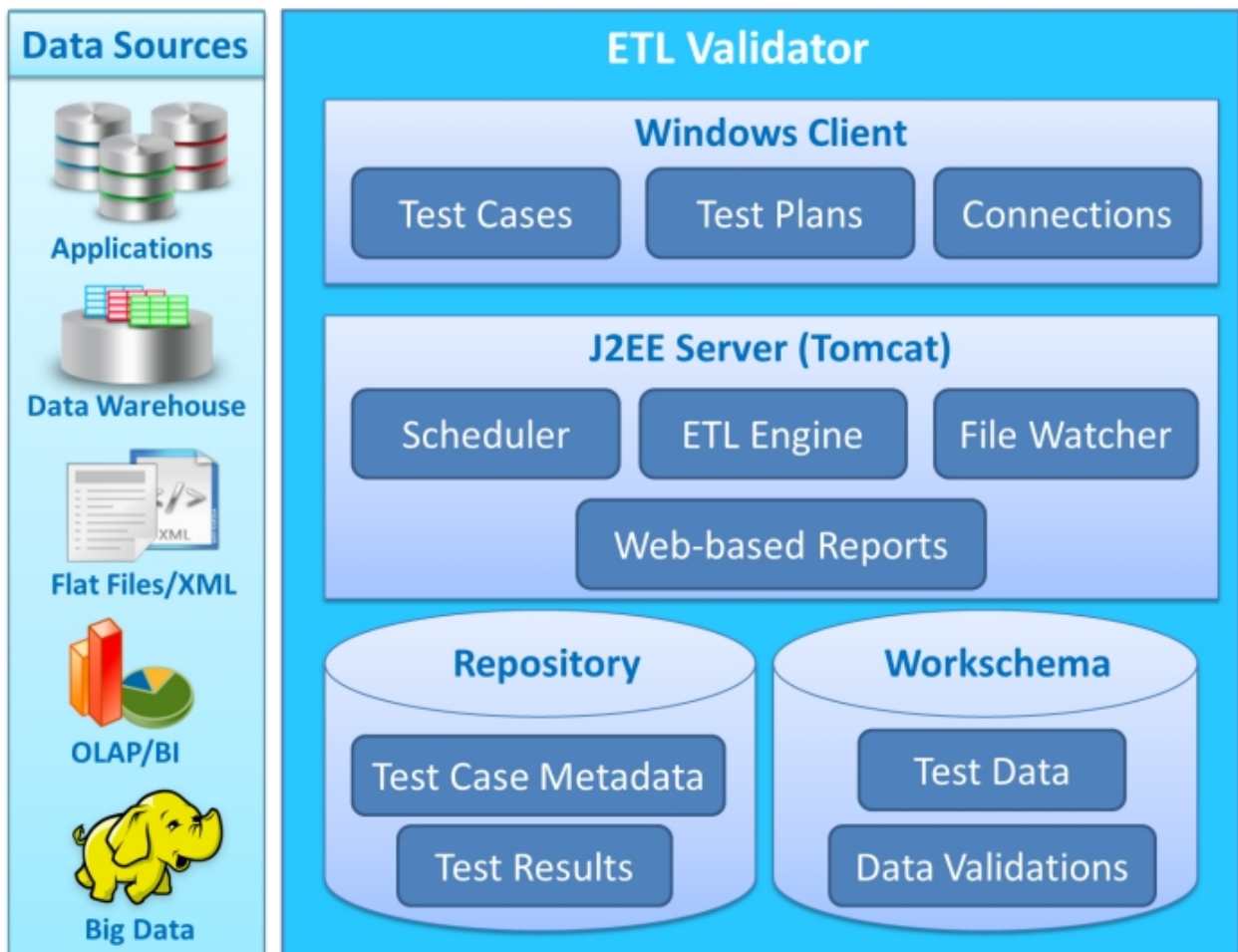
[Client-Server Architecture](#)

[ETL Validator Topology](#)

Client-Server Architecture

ETL Validator Client-Server Architecture

ETL Validator has a Client-Server Architecture with a Repository database for story test case metadata and a Workschema database for storing the test data.



The key components of ETL Validator are :

ETL Validator Client

ETL Validator Client is a windows based application that provides the User Interface for creating Test Cases, Test Plans and managing their execution in the ETL Validator Server. ETL Validator Client also provides interface for administration of the solution.

ETL Validator ELV Server

ETL Validator ELV server is a J2EE application running in Tomcat used for executing of Test Cases and Test Plans. It has a built in ETL engine for extracting and loading test data from heterogeneous data sources. It also has a scheduler for automatic execution of Test Plans at specified times and a file watcher that checks for any new files arriving in the specified folder before kicking off Test Plans.

ETL Validator Repository

ETL Validator Repository stores the metadata related to test cases, test plans, data source connections and users. It requires an Oracle or Postgresql database.

ETL Validator Work Schema

ETL Validator Work Schema stores the test data, benchmark data and the data validation results. It requires an Oracle or Postgresql database.

ETL Validator Reports

ETL Validator Reports is a web based interface for viewing Test Case and Test Plan run results. In addition, it allows the user to download results in a PDF or Excel format. It is a Java based application that can be deployed in Tomcat along with the ETL Validator Server.

ETL Validator Command Line

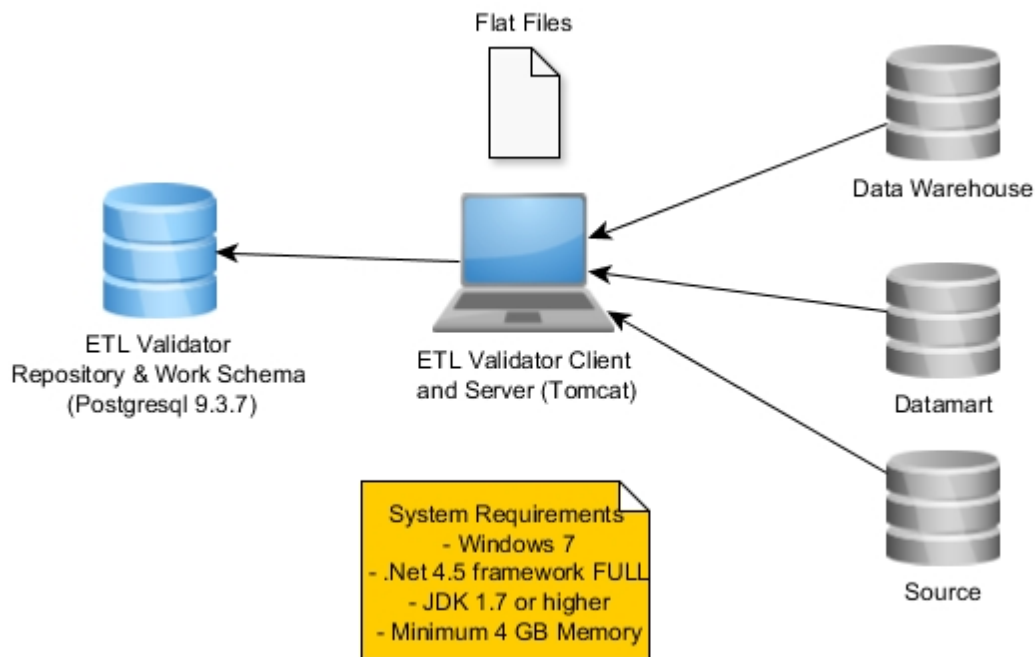
A jar file (dgcl.jar) for remotely invoking test plan and test case runs from the command line. Additionally, it provides a command interface for invoking test case from HP Application Lifecycle Manager (ALM). More details about the available command line interface can be found on our support site [here](#).

Topology

ETL Validator Topology

ETL Validator can be implemented in the following two topologies:

- **ETL Validator Complete (Single user)** - Typically used by a single user, this topology consists of ETL Validator Client and Server running on the same windows 7 computer while the repository is a Postgresql database. It can be setup by running the installer available in the "ETL Validator Complete" download file. The installer automatically installs the ETL Validator Client, the Postgresql database for the repository / workschema and the server running on tomcat7. This configuration does require JDK 1.7 or later and .Net 4.5 Framework Full. The embedded Postgresql database also includes a schema with sample data.



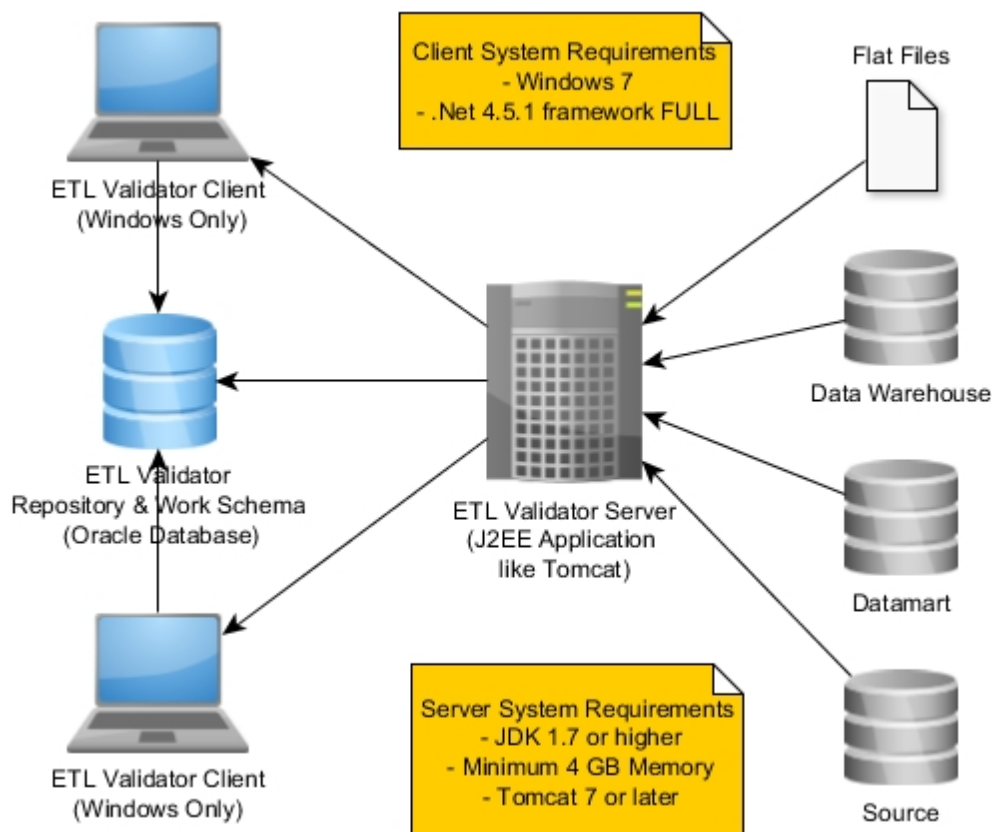
Ports used by ETL Validator Complete:

Default Port Number for Tomcat 7 or later for running ETL Validator Server and ETL Validator Reports : 6050

Default Port Number for Postgresql database which stores the Repository/
Workschema : 6060

Default Port Number for pgbouncer service used to access the Postgresql database:
6070

- **ETL Validator for multi-user environment :** In this topology, the ETL Validator Client and Server are installed separately on different computers using "ETL Validator Client Only" and "ETL Validator Server" install files. The server can be deployed in any J2EE application server but Tomcat 7 or later is recommended. The server also requires JDK 1.7 or later. Oracle database is typically used as the Repository/Workschema database deployed centrally for all the users to connect.



ETL Validator Setup

[System Requirements](#)

[Database Requirements](#)

[Configuration Wizard](#)

[Repository Setup using ETL Validator Client](#)

[Workschema Setup using ETL Validator Client](#)

[DB Connection Setup](#)

[BI Connection Setup](#)

[Usage Tracking DB Setup](#)

[Certification Matrix](#)

System requirements

ETL Validator Client and Server system requirements

a) Server Requirements: User must have the below requirements in the server side>>

- Java 1.7 or above
- J2EE Application server
 - Tomcat 7 or above
- Dual-core processor minimum ~ Recommended: Quad-core processor or better
- 4 GB RAM minimum ~ Recommended: 8 GB or more
- 20 GB of hard-disk space for program installation and operation

b) Client Requirements: User must have the below requirements in the client side>>

- Microsoft .Net Framework 4.5.1 : <http://www.microsoft.com/en-us/download/details.aspx?id=40773>
- Windows XP or Windows 7 operating system
- Dual-core processor minimum ~ Recommended: Quad-core processor or better
- 2 GB RAM minimum ~ Recommended: 4 GB or more
- 2 GB of hard-disk space for program installation and operation
- Display dimensions of 1024x768 or greater

Related : [Database Requirements](#)

Database Requirements

ETL Validator requires Oracle 11g or Postgresql 9.3 for [creating a Repository and Workschema](#) . ETL Validator Repository and Workschema cannot be colocated in the same database user. So two database users are needed >>

- The database user must have the following privileges for Repository creation. A sample script for creating the database user "qprobe" in Oracle 11g is shown below>>

```
CREATE USER qProbe IDENTIFIED BY qprobe;
```

```
GRANT CONNECT, RESOURCE, CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE  
SEQUENCE, CREATE PROCEDURE, CREATE TYPE, CREATE SYNONYM TO qprobe;
```



```
ALTER USER qprobe DEFAULT ROLE NONE;
```

Note: 'Create Table' privilege should be granted explicitly to the DB user account and not through a role as shown in the above script.

Recommended table space is 20 GB.

- Same privileges are needed for the creating Workschema. A sample script for creating the database user "qprobews" in Oracle 11g is shown below >>

```
--ETL Validator Workschema (Ex: qprobews)
```

```
CREATE USER qProbews IDENTIFIED BY qprobews; GRANT CONNECT, RESOURCE,
      CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SEQUENCE, CREATE
      PROCEDURE, CREATE TYPE, CREATE SYNONYM TO qprobews;
```

```
ALTER USER qprobews DEFAULT ROLE NONE;
```

Recommended table space is 30 GB but this may vary depending on the number of test cases and the volume of the data being compared.

Server Setup

Below are the steps for setting up ETL Validator Server

Step 1 : Install JDK 1.7 or later

JDK can be downloaded from the oracle website : <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Step 2 : Install Tomcat 7 or later

Tomcat can be downloaded and installed from the apache.org website : <http://tomcat.apache.org/download-70.cgi>

Create admin users in tomcat that can be used to login from the tomcat manager app. Users can be configured by editing the file : <tomcat install dir>/conf/tomcat-users.xml

Add an entry for the tomcat admin user.

```
<tomcat-users>
<user roles="manager,manager-ui,manager-gui,admin-gui" password="admin" username="admin"/>
</tomcat-users>
```

Verify that the java_home has been setup and start tomcat.

Step 3 : Deploy ETL Validator Server and ETL Validator Service (reporting service) in Tomcat

Download ETL Validator sever using the download URL provided by your datagaps support team and extract it to a local folder.

Login to the tomcat manager app by clicking on the 'Manager App' as shown below using the admin user created in the previous step :

Apache Tomcat/7.0.57

If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:

[Security Considerations HOW-TO](#)

[Manager Application HOW-TO](#)

[Clustering/Session Replication HOW-TO](#)

[Server Status](#)
[Manager App](#)
[Host Manager](#)

Deploy ETLValidatorServer.war file from datagaps server download using the deployment option in tomcat manager app as shown below :

Deploy	
Deploy directory or WAR file located on server	
Context Path (required):	<input type="text"/>
XML Configuration file URL:	<input type="text"/>
WAR or Directory URL:	<input type="text"/>
<input type="button" value="Deploy"/>	
WAR file to deploy	
Select WAR file to upload	<input type="button" value="Choose File"/> No file chosen <input type="button" value="Deploy"/>

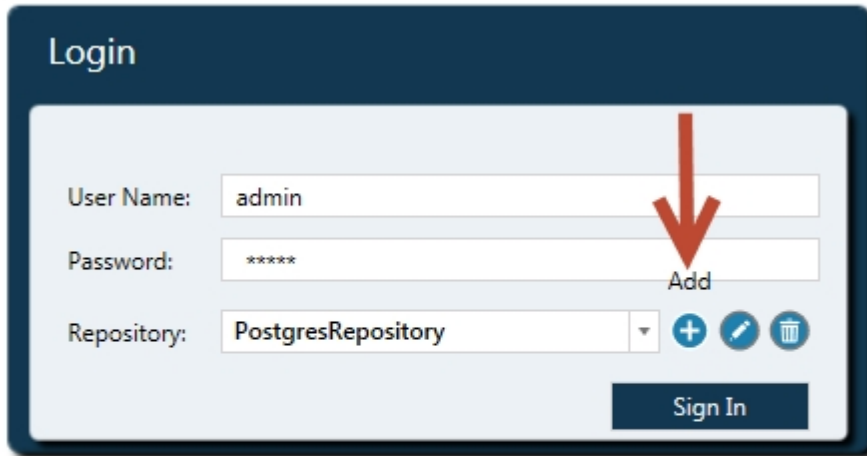
Verify that the ETL Validator Server got install successfully and is running by clicking on the deploy application link as shown below. It should show a page with the ETL Validator Server version.

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes
/ETLValidatorServer	None specified		true	0	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ 30 minutes

Once the server setup is completed, please restart the ETL Validator Server (eg. tomcat) for the updated repository settings to take effect.

Client and Repository/Workschem Setup

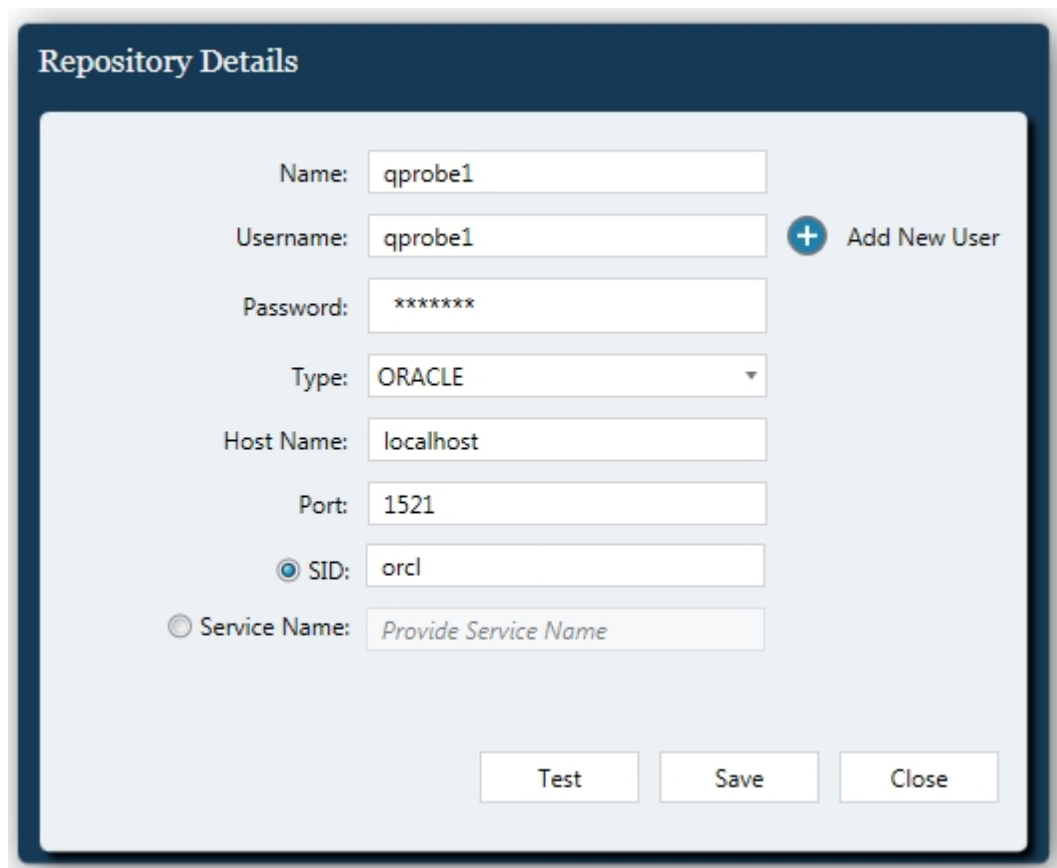
Step 1: Add a new repository by clicking on the 'New' button.



The screenshot shows the ETL Validator Login screen. It features a dark blue header with the text "Login". Below the header is a light blue form area. The form contains three input fields: "User Name" with the value "admin", "Password" with the value "*****", and "Repository" with a dropdown menu showing "PostgresRepository". To the right of the "Repository" dropdown are three circular icons: a plus sign, a pencil, and a trash can. A red arrow points down to the plus icon. Below the icons is a dark blue "Sign In" button.

ETL Validator comes with an embedded repository database called 'PostgresRepository' for saving Test Cases and Test Plans. The default username/ password are admin/ admin. Please click on 'Sign In' button to start using ETL Validator.

Step 2: Provide repository database user details. ETL Validator creates the repository using these details

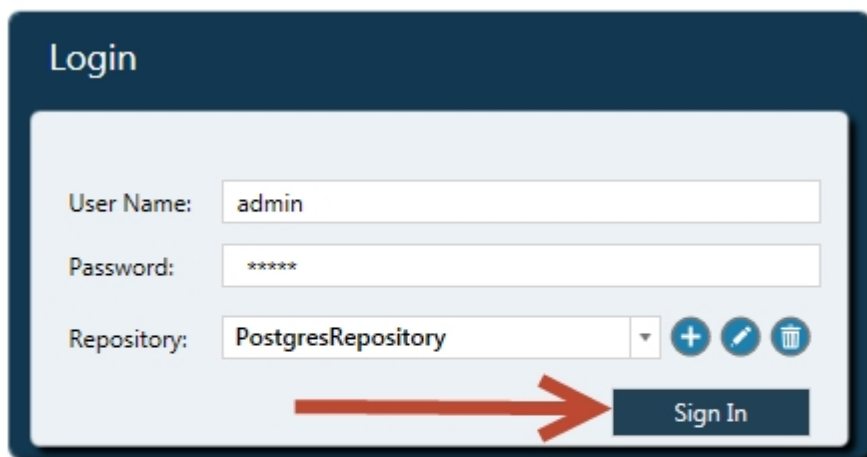


The 'Repository Details' dialog box contains the following fields and controls:

- Name:
- Username: + Add New User
- Password:
- Type:
- Host Name:
- Port:
- ☒ SID:
- ☐ Service Name:
- Buttons: Test, Save, Close

Click on 'Test' to verify the connection. Once the connection has been verified successfully, click on 'Save' to save. If the repository is blank or has tables from a previous version, the ETL Validator Client automatically prompts for approval to install/upgrade the repository.

Step 3: Signin



The 'Login' dialog box contains the following fields and controls:

- User Name:
- Password:
- Repository: + ✎ 🗑
- Sign In button (indicated by a red arrow)

ETL Validator comes with an embedded repository database called 'PostgresRepository' for saving Test Cases and Test Plans. The default username/ password are admin/ admin. Please click on 'Sign In' button to start using ETL Validator.

Step 4: Specify ETL Validator Server URL

Configuration

- Repository Database
- Repository Privileges
- Repository Version
- Server Accessibility**
- Server Repository Connection
- Workschema Checks

Server Settings

ETL Validator Server URL :

Path for temp files:

ETL Validator server Url Verified Successfully.

Test **Save**

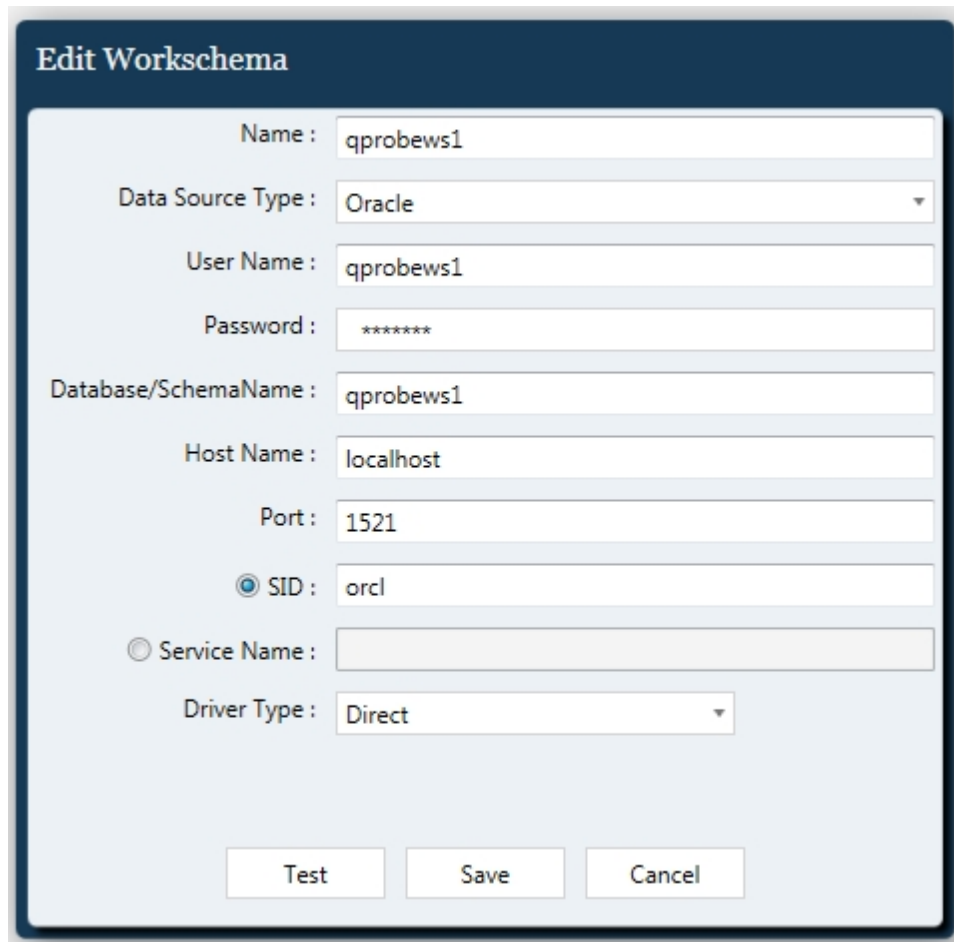
Provide the server URL and click on 'Test'. The server URL format should be :
 http://<sever host machine name>:<tomcat port eg.8080>/ETLValidatorServer/Execute?wsdl

The following two checks happen when the 'Test' button is clicked :

1. Access to the ETL Validator Server is checked
2. ETL Validator server checks if it is able to connect to the repository

Save the server setting and restart the tomcat in the server machine so that the ETL Validator Server repository connection details are applied.

Step 5: Provide workschema database user details. ETL Validator creates the workschema using these details



Edit Workschema

Name : qprobews1

Data Source Type : Oracle

User Name : qprobews1

Password : *****

Database/SchemaName : qprobews1

Host Name : localhost

Port : 1521

☒ SID : orcl

☐ Service Name :

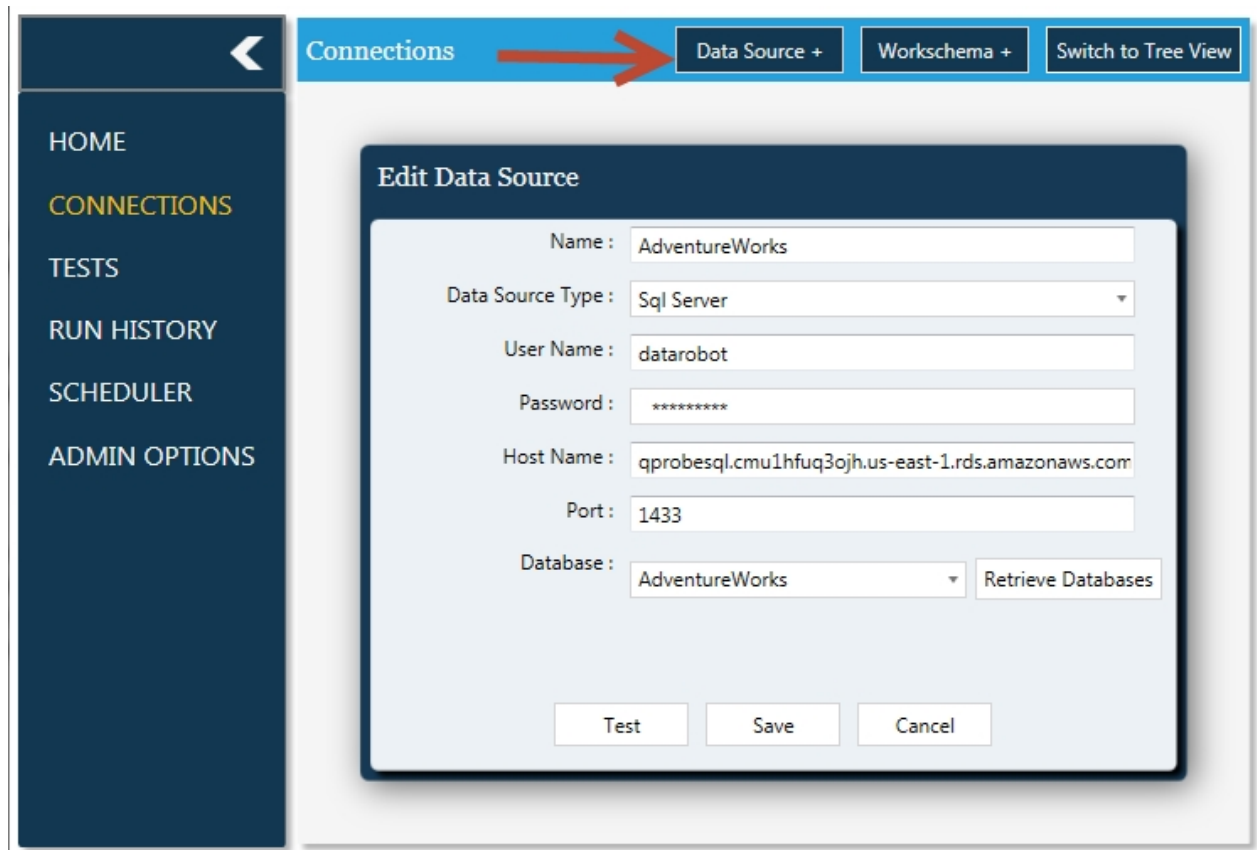
Driver Type : Direct

Test Save Cancel

'Test' the connection and 'Save' it. ETL Validator will prompt for the creation of the 'Workschema'. Once complete, ETL Validator home screen will be displayed.

Connection Setup

Create Connections to datasources (Connections for source, Stage, Target databases should be specified here)



Certification Matrix

ETL Validator supports connection to the following Database Servers.

Server	Version
Oracle	10g or above
SQL Server	2000 or above
Teradata	13.0 or above
IBM DB2	9.7 or above
MySql	5.5 or above
Netezza	7.2 or above
Vertica	7.1 or above
Postgresql	9.1 or above
Sybase	Coming soon
Hadoop Hive	2

ETL Validator supports sourcing of data from the following Reporting Tools.

Tool	Version
OBIEE	11g

Replicating ETL Validator Setup

Assumption : You have an existing ETL Validator repository (eg. qprobe), workschema (eg. qprobews) and server that you are interested in replicating to a new environment or create a copy of it in the same environment so that you can have multiple instances of ETL Validator.

There are three steps:

1. Create a copy of the repository and work schema
2. Setup ETL Validator Server
3. Connect to the new repository using ETL Validator Client and configure the Server

Copy Repository and Workschema

1. Create two new oracle 11g database users - one for the repository and one for the workschema.
2. Make sure these database users have the following privileges (eg. for qprobe) :

```
GRANT CONNECT, RESOURCE, CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SEQUENCE, CREATE PROCEDURE, CREATE TYPE, CREATE SYNONYM TO qprobe;
```

```
ALTER USER qprobe DEFAULT ROLE NONE;
```

See [Database Requirements](#)

3. Export and import the database schemas for your existing repository (qprobe) and workschema (qprobews) to the newly created repository and workschema users.

Setup ETL Validator Server

1. Download and install Tomcat from <http://tomcat.apache.org/>
2. If you are planning to run multiple ETL Validator Servers in the same machine, you can setup multiple instances of tomcat in the same server. Detailed instructions can be found in the tomcat manual : <http://tomcat.apache.org/tomcat-7.0-doc/RUNNING.txt>

(1) The most common hiccup is when another web server (or any process for that matter) has laid claim to port 8080. This is the default HTTP port that Tomcat attempts to bind to at startup. To change this, open the file:

\$CATALINA_HOME/conf/server.xml

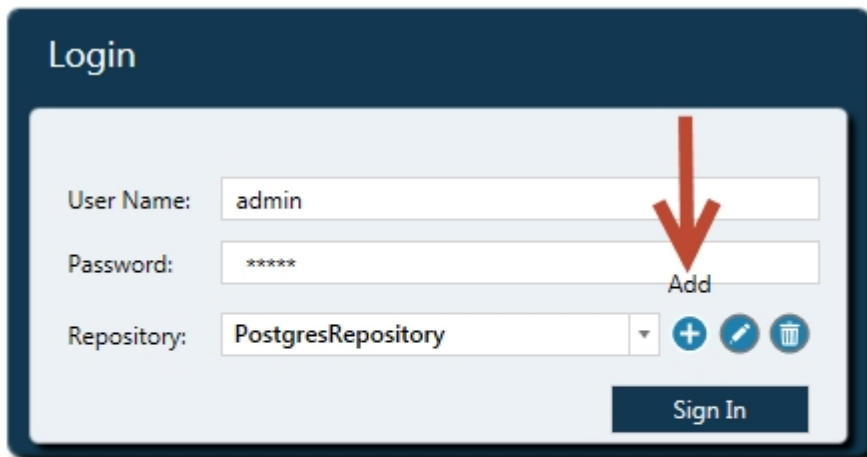
and search for '8080'. Change it to a port that isn't in use, and is greater than 1024, as ports less than or equal to 1024 require superuser access to bind under UNIX. Restart Tomcat and you're in business. Be sure that you replace the "8080" in the URL you're using to access Tomcat. For example, if you change the port to 1977, you would request the URL <http://localhost:1977/> in your browser.

(2) Modify the AJP Connector port from 8009 to an available port number greater than 1024.

3. Deploy ETLValidatorServer.war file

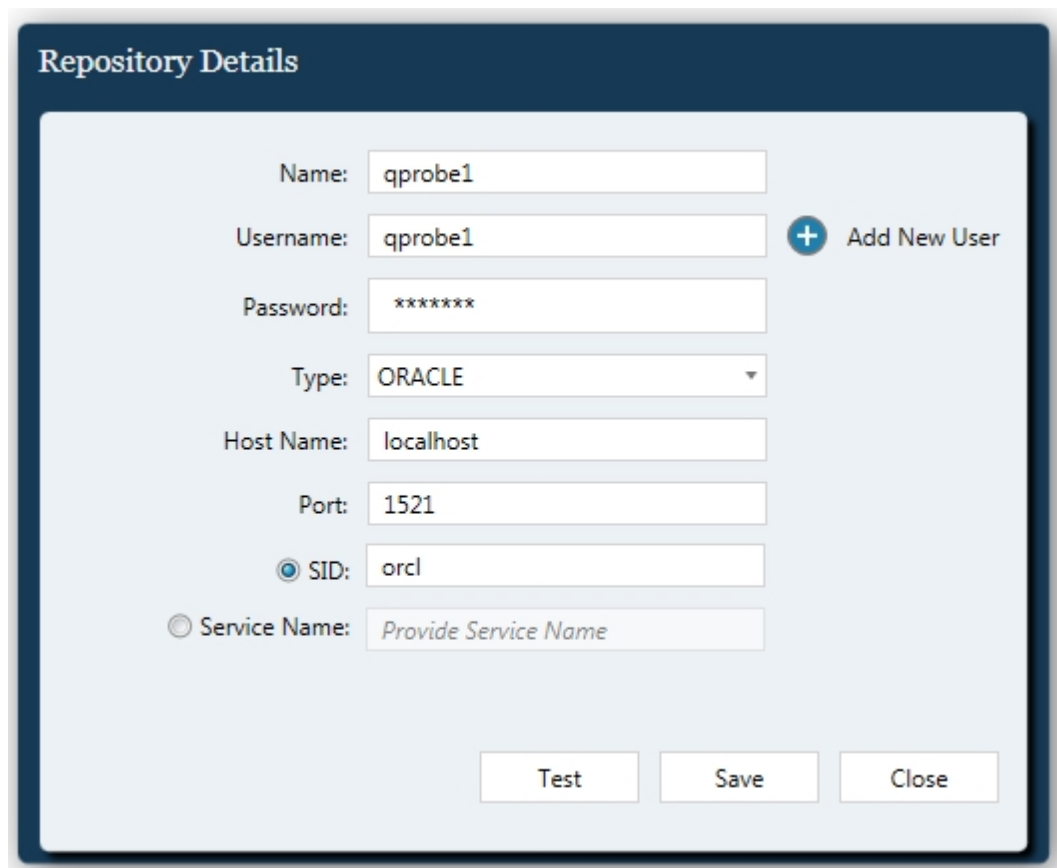
Configuring Client to new Repository

Step 1: Add a new repository by clicking on the 'New' button.



ETL Validator comes with an embedded repository database called 'PostgresRepository' for saving Test Cases and Test Plans. The default username/ password are admin/ admin. Please click on 'Sign In' button to start using ETL Validator.

Step 2: Provide repository database user details. ETL Validator creates the repository using these details

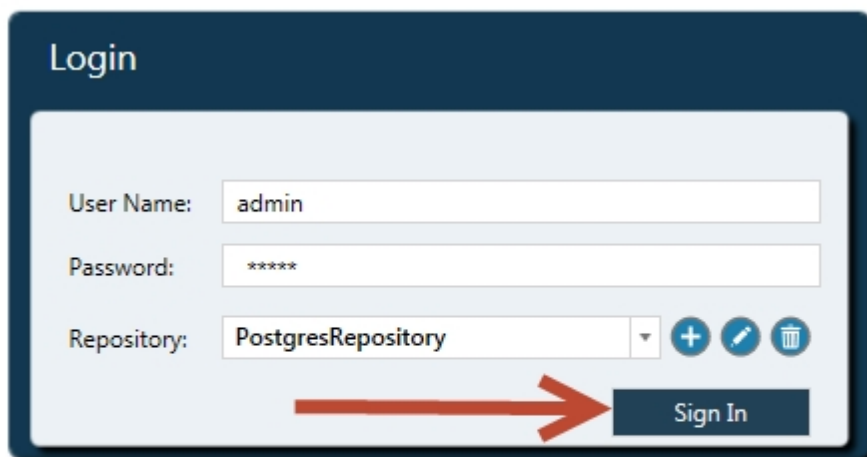


The 'Repository Details' dialog box contains the following fields and controls:

- Name:
- Username: + Add New User
- Password:
- Type:
- Host Name:
- Port:
- ☒ SID:
- ☐ Service Name:
- Buttons: Test, Save, Close

Click on 'Test' to verify the connection. Once the connection has been verified successfully, click on 'Save' to save. If the repository is blank or has tables from a previous version, the ETL Validator Client automatically prompts for approval to install/upgrade the repository.

Step 3: Signin



The 'Login' dialog box contains the following fields and controls:

- User Name:
- Password:
- Repository: + ✎ 🗑
- Sign In button (indicated by a red arrow)

ETL Validator comes with an embedded repository database called 'PostgresRepository' for saving Test Cases and Test Plans. The default username/ password are admin/ admin. Please click on 'Sign In' button to start using ETL Validator.

Step 4: Specify ETL Validator Server URL

The screenshot shows the 'Configuration' window with a sidebar on the left and a main panel on the right. The sidebar contains a list of configuration items, each with a colored circle: 'Repository Database' (green), 'Repository Privileges' (green), 'Repository Version' (green), 'Server Accessibility' (red), 'Server Repository Connection' (grey), and 'Workschema Checks' (grey). A blue arrow points from the 'Server Accessibility' item to the main panel. The main panel has a blue header 'Server Settings' with a green circle on the right. It contains two text input fields: 'ETL Validator Server URL : http://localhost:6050/ETLValidatorServer/Execute?wsdl' and 'Path for temp files: Path to save ETL files'. Below these fields is a blue message 'ETL Validator server Url Verified Successfully.' and two dark blue buttons labeled 'Test' and 'Save'.

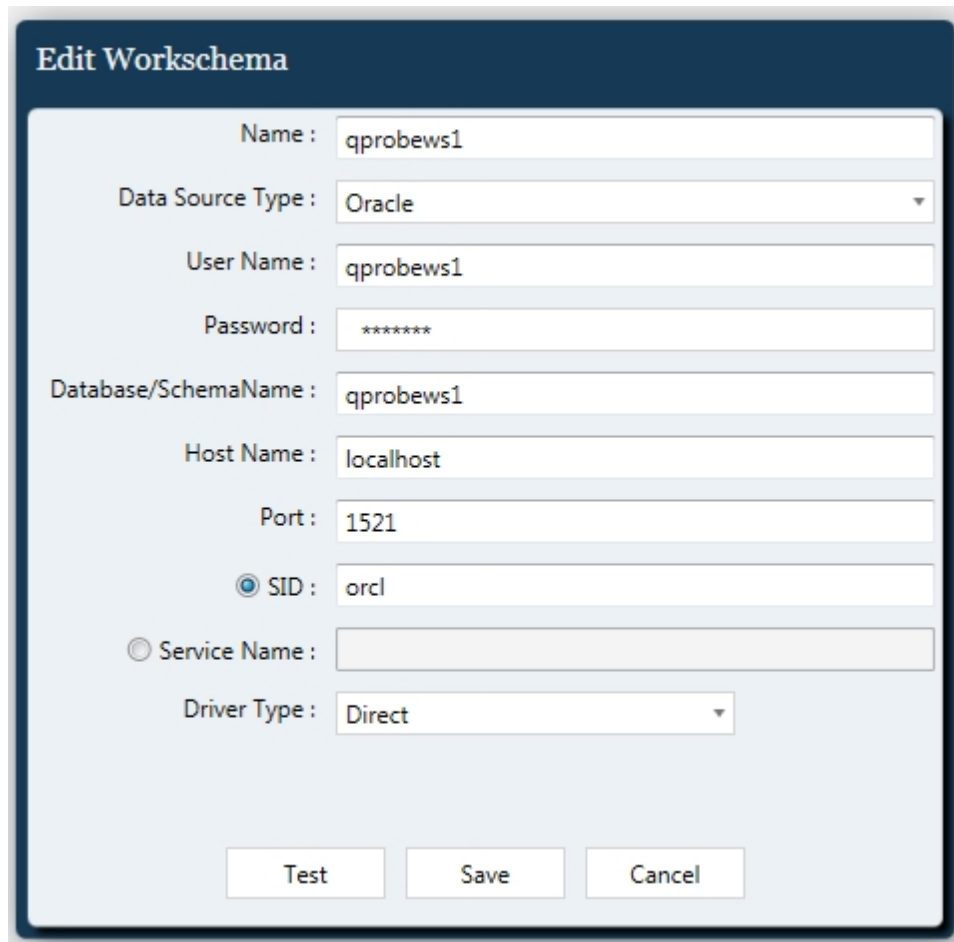
Provide the server URL and click on 'Test'. The server URL format should be :
`http://<sever host machine name>:<tomcat port eg.8080>/ETLValidatorServer/Execute?wsdl`

The following two checks happen when the 'Test' button is clicked :

1. Access to the ETL Validator Server is checked
2. ETL Validator server checks if it is able to connect to the repository

Save the server setting and restart the tomcat in the server machine so that the ETL Validator Server repository connection details are applied.

Step 5: Provide workschema database user details. ETL Validator creates the workschema using these details



Edit Workschema

Name : qprobews1

Data Source Type : Oracle ▼

User Name : qprobews1

Password : *****

Database/SchemaName : qprobews1

Host Name : localhost

Port : 1521

☒ SID : orcl

☐ Service Name :

Driver Type : Direct ▼

Test Save Cancel

'Test' the connection and 'Save' it. ETL Validator will prompt for the creation of the 'Workschema'. Once complete, ETL Validator home screen will be displayed.

Copyright © 2010, 2015, datagaps inc. All rights reserved.