



Hewlett Packard Enterprise Trueview

Installation Guide

Release 8.0

First Edition



Hewlett Packard
Enterprise

Notices

Legal notice

© Copyright 2016 Hewlett Packard Enterprise Development LP

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the US

Trademarks

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Contents

Notices	1
Preface	5
1.1 About this guide.....	5
1.2 Document history.....	5
Chapter 2 Trueview Installation.....	6
2.1 Distribution.....	6
2.2 Install Trueview Files.....	6
2.3 Create Trueview Database.....	8
2.3.1 Oracle JDBC Driver Files.....	8
2.3.2 Oracle Database.....	9
2.3.3 PostgreSQL Database.....	11
2.3.3.1 PostgreSQL Installation	11
2.3.4 Setup TNP logging	14
2.3.5 Trueview Database Application Configuration	15
2.4 Configuring Trueview on a Web Server	15
2.4.1 WebLogic Configuration and Startup.....	15
2.4.2 Jboss Installation, Configuration, and Startup.....	25
2.4.2.1 JBoss Installation.....	25
2.4.2.2 JBoss (TNP) Start/Status/Stop.....	30
2.4.2.3 Trueview Installation	30
2.4.2.4 JBoss Configuration	31
2.4.2.5 Trueview Startup/Shutdown.....	34
2.5 Configuring Trueview for Event Collection	35
2.5.1 Collector Configuration.....	35
2.5.2 Collector Wrapper Scripts.....	35
2.6 Configuring Connectors.....	36
2.6.1 Alcatel SAM Connector.....	36
2.6.2 ANDA Connector.....	37
Chapter 3 Trueview Upgrade.....	38
3.1 Install Trueview Files	38
3.2 Update the Deployed Applications	42
3.2.1 WebLogic.....	42
3.2.2 JBoss.....	43
Chapter 4 Verification of signed binary.....	45
4.1 Installing and Configuring Gnu Privacy Guard (GnuGP).....	45
4.2 Verifying the Authenticity and Integrity of the Software.....	45

List of tables

Table 1: Document history	5
---------------------------------	---

List of figures

Figure 1 System Configuration.....	15
Figure 2 WebLogic Administration Console.....	17
Figure 3 WebLogic configuration example.....	20
Figure 4 WebLogic deployments.....	43
Figure 5 WebLogic update application assistant.....	43

Preface

1.1 About this guide

The purpose of this guide is to explain the steps for installing and configuring Trueview.

1.2 Document history

Table 1: Document history

Edition	Date	Description
1.0	31 March 2016	First official version.

Chapter 2

Trueview Installation

The following instructions go through the steps to install Trueview. It is assumed that a supported web server (example: WebLogic 12.1.2 or RedHat JBoss 6.2.1) is already installed and a supported database product (e.g. Oracle 12c or Postgres 9.1) is also installed.

2.1 Distribution

The top-level folder on the CD or zipped install file is in the following format:

```
<Dir>    tnp-<version#>-<build#>
<Dir>    ProductDocumentation
```

The `tnp-<version#>-<build#>` directory has the following subdirectories:

```
<Dir>    adapters
<Dir>    api
<Dir>    bin
<File>    BUILD_ID
<Dir>    connectors
<Dir>    database_scripts
<Dir>    distrib
<File>    docs.zip
<Dir>    eventhandling
<Dir>    lib
<Dir>    NetSmartConnector
<Dir>    prov-configuration
<File>    README.txt
<Dir>    reports
<File>    samples.zip
<Dir>    svg
<Dir>    utils
```

2.2 Install Trueview Files



NOTE: Each step in this section assumes the previous steps have been executed. Therefore, skipping a step may lead to different and unexpected results.

1. Create a Unix account called **Trueview** on the application server, with group permissions set for readability by the installed web server user (see section 2.4).

```
$ useradd -c <comment> -d <Trueview_homDir> -g <group> -m -s <shell> Trueview
```

For example:

```
$ useradd -c "Trueview owner" -d /home/Trueview -g staff -m -s /bin/csh Trueview
```



NOTE: <Trueview_homeDir> will represent Trueview's home directory throughout this guide. In this example <Trueview_homeDir> is /home/Trueview.

2. Log into the Unix-based server using the Trueview user account and execute the following commands:

```
$ cd
$ chmod 750 ./
$ mkdir app
$ cd app
```

3. Copy the application release contents from the physically shipped CD or extract the content from the digitally shipped zipped file into the previously created Trueview app directory. There are a number of files and directories included in the distribution. Please ensure that all files are transferred and/or extracted.

If physical CD, first place the CD into a reachable device for the host to read, then execute:

```
$ scp -pr <CD>:* ./
```

If digital file, first transfer the file(s) into the current directory, then execute:

```
$ gunzip <digitalfile>.zip
$ tar xf *.tar
```

Upon completion there will be some or all of these directories:

- tnp-<version#>-<build#>
- CustomEventHandlers
- JBoss
- JBossTNP
- custom

4. Log out of the Trueview user account.
5. Log in as the web server user (e.g. weblogic or jboss). Trueview needs to be configured via the web server user account and not the Trueview user account (Trueview).



NOTE: <webserver_homeDir> will represent web server's home directory throughout this guide. For example <webserver_homeDir> could be /home/weblogic.

```
$ cd
$ mkdir tnp-config
$ cd tnp-config
$ mkdir <tnpwebobjectname>
$ cd <tnpwebobjectname>
$ ln -s <Trueview_tarDir>/<tnpBuild> release
```



NOTE: For clarity, the <tnpwebobjectname> usually contains the web server name being used. It is recommended that you refer to the appropriate web server section before deciding on a name. <Trueview_tarDir>/<tnpBuild> refers to the location where the distribution was placed in the previous steps, either from CD or from the digital file.

6. Create links to the release directory so that the web server can easily find the most recent software files. This is done to reduce the amount of effort required should there be need to upgrade the existing environment later:


```
$ ln -s release/* ./
```

- It is necessary for the web server user (weblogic or jboss) to have read-write permission to the adapters and reports directories. In order to achieve this, the links are removed and the entire contents of the directories are copied over.

```
$ rm adapters
$ cp -rp release/adapters ./

$ rm reports
$ cp -rp release/reports ./
```

- It is necessary for the web server user to have read-write permission to the custom eventhandling directory. In order to achieve this, the link is removed and the entire contents of the eventhandling directory is copied over. In addition, all custom handlers (if any) are also copied in:

```
$ rm eventhandling
$ cp -rp release/eventhandling ./
$ cp -p <Trueview_tarDir>/CustomEventHandlers/* eventhandling/
```

- A valid license is required to run Trueview. For security reasons, the license file <tnplicense> is sent as a separate password-protected file and is not included in the distribution CD.



NOTE: Please contact Support Services if you have not received a license file.

First, copy the received license file into the previously created **Trueview** app directory (<Trueview_homeDir>/app).

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>
$ mkdir license
$ cd license
$ cp <Trueview_homeDir>/app/<tnplicense> ./
```

2.3 Create Trueview Database

There are two databases that comprise the complete Trueview application. As a minimum, there is the main TNP database. If the environment is making use of Event Management, there is a second database referred to as EVM.

2.3.1 Oracle JDBC Driver Files

When installing Trueview the following Oracle JDBC driver files:

```
ojdbc6.jar
orai18n.jar
```

Must be downloaded from the Oracle website:

<http://www.oracle.com/technetwork/apps-tech/jdbc-112010-090769.html>

To download the files, the Oracle license terms must be accepted.

These files must be placed in the following Trueview directory: [ReleaseDirectory]/lib

2.3.2 Oracle Database

A shell script is provided as part of the distribution to manage creation of the new database tablespace schema as well as updates to the existing schema.

1. Install Oracle 12g. Refer to Oracle 12g installation Guide.
2. Create auto extend tablespace named <dbtablespaceTNP>. Refer to Oracle 11g Documentation. If EVM is installed, create auto extend tablespace named <dbtablespaceEVM>
3. Log into database and create a Trueview database user account with the following parameters:

```
Username = <dbusernameTNP>
Password = <dbpasswordTNP>
Grant Role = CONNECT, RESOURCE
System Grants = UNLIMITED_TABLESPACE, CREATE VIEW, CREATE SYNONYM, CREATE JOB,
CREATE EXTERNAL JOB
Table space name = <dbtablespaceTNP>
```



NOTE: It is assumed that the privilege of DBA will not be granted. If DBA will be allowed, then the Role and Grants can be changed to CONNECT, DBA, EXP_FULL_DATABASE and IMP_FULL_DATABASE.



NOTE: It is possible to confirm the user account settings via the SQL statement:
select * from SESSION_PRIVS.

If the environment is making use of Event Management then a second account needs to be created in the same fashion as above:

```
Username = <dbusernameEVM>
Password = <dbpasswordEVM>
Grant Role = CONNECT, RESOURCE
System Grants = UNLIMITED_TABLESPACE, CREATE VIEW, CREATE SYNONYM, CREATE JOB,
CREATE EXTERNAL JOB
Table space name = <dbtablespaceEVM>
```

4. As the web server user, execute the Trueview database update script to add the required tables to the created database:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>/release/database_scripts
$ ./databaseUpdate.sh
```

You will be prompted to enter specific parameters. The following is the console output of the script and the required responses:

```
Do you want to continue? (y/n)
y
Please enter the database type for this installation (oracle, postgres):
oracle
Please enter database user name:
<dbusernameTNP>
Please enter database user password:
<dbpasswordTNP>
Please enter the database server host name or ip:
<localhost> or <IPaddr> or <hostname>
```

```

Please enter the database server port(1521):
<dbserverport>
Please enter the Oracle database SID:
<SID>
Please enter the tablespace name for data:
<dbtablespaceTNP>
Please enter the tablespace name for indexes:
<dbtablespaceTNP>
Please enter the LDAP authentication scheme for this installation (openldap,
activedirectory): openldap

```

The script will finish with successful messages such as the following:

```

Updating Change Set Checksums ...
Migration successful
Updating Application Schema ...
Migration successful
Updating Application Privileges ...
Migration successful

```



NOTE: Since all required Trueview database tables are being created by these scripts, the total running time might be close to 2-4 minutes. Confirm that three “Migration successful” messages are returned from the update script as it runs. If any one step is unsuccessful, the other will not run – contact HEWLETT PACKARD ENTERPRISE Support for assistance with any error messages on installation/configuration.

5. If the environment is making use of Event Management, then as the web server user, execute the Trueview EVM database update script to add the required tables to the created database:

```

$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>/database_scripts
$ ./evm-databaseUpdate.sh

```

You will be prompted to enter specific parameters. The following is the console output of the script and the required responses:

```

Do you want to continue? (y/n)
y
Please enter the database type for this installation (oracle, postgres):
oracle
Please enter database user name:
<dbusernameEVM>
Please enter database user password:
<dbpasswordEVM>
Please enter the database server host name or ip:
<localhost> or <IPaddr> or <hostname>
Please enter the database server port(1521):
<dbserverport>
Please enter the Oracle database SID:
<SID>
Please enter the tablespace name for data:
<dbtablespaceEVM>
Please enter the tablespace name for indexes:
<dbtablespaceEVM>
Please enter the LDAP authentication scheme for this installation (openldap,
activedirectory): openldap

```

The script will finish with successful messages such as the following:

```

Updating Application Schema ...

```

```
Migration successful
Updating Application Privileges ...
Migration successful
```



NOTE: Since all required Trueview database tables are being created by these scripts, the total running time might be close to 2-4 minutes. Confirm that one "Migration successful" message is returned from the update script as it runs. If any one step is unsuccessful, the other will not run – contact HEWLETT PACKARD ENTERPRISE Support for assistance with any error messages on installation/configuration.

2.3.3 PostgreSQL Database

PostgreSQL 9.1.2 is supported by Trueview 8.0.

2.3.3.1 PostgreSQL Installation

1. User Account Setup
 - a. As root, create user 'postgres'

```
useradd -c "Postgres Admin" -d /home/postgres -g users -m -s /bin/bash postgres
passwd postgres
```

- b. Set environment variables in `~postgres/.bash_profile`

```
PG_HOME=/opt/postgres/9.1
PGDATA=$PG_HOME/data
PATH=$PATH:$HOME/bin:$PG_HOME/bin
export PG_HOME PGDATA PATH
```

2. Installation and Startup
 - a. Download Postgres Enterprise Manager from EnterpriseDB (<http://www.enterprisedb.com/products-services-training/pgdownload>) to `host : /opt/TAR/ postgresql-9.1.19-3-linux-x64.run`
 - b. As root, run the install file which automatically starts PG after installation

```
./postgresql-9.1.19-3-linux-x64.run
```



NOTE: The RPM install script prompts for "Specify superuser password [password]:" This is NOT the password for superuser ROOT -- it is what you want to set as the password for PostgreSQL DATABASE superuser. This password will be required to connect to the PostgreSQL server. Provide preferred password or accept the default: "password".

Most of the `/opt/postgres/9.1/` subdirectories (e.g. libraries, binaries, etc.) remain under *root* ownership to protect against accidental change or deletion. The `data/` subdirectory is owned by the postgres user.

- c. Add entry(s) in `$PG_HOME/data/pg_hba.conf` for remote db access

```
host    all             all             remote_IP/32    md5
```

- d. Make change(s) take effect

```
pg_ctl reload
```

3. Setup Postgres DB for TNP
 - a. Create a db user 'tnp94'
 - i. As postgres,

```
$PG_HOME/bin/createuser --superuser --createdb --encrypted --pwprompt tnp94
```

- b. Create a directory for db tablespace 'tnp94'
 - i. As postgres,

```
cd $PG_HOME/data/pg_tblspc
mkdir tnp94
```

- c. Download pgAdmin3 client from <http://www.pgadmin.org/download/>
- d. Open pgAdmin3 client and "Add a new connection to a server". Fill in the following fields:
 - i. Name field: <dbservername> or <dbserverip>
 - ii. Port : <dbserverport>



NOTE: Default value is 5432

- iii. Maintenance DB: postgres
- iv. Username: postgres



NOTE: Password is added later during installation.

Upon completion, click **OK**.

- e. From pgAdmin3 GUI, create a 'tnp94' tablespace

```
Name=tnp94
Owner=tnp94
Location=$PG_HOME/data/pg_tblspc/tnp94
```

- f. Create db 'tnp94'
 - As postgres,

```
$PG_HOME/bin/createdb tnp94 --owner=tnp94 --tablespace=tnp94
```

4. Setup Postgres DB for EVM
 - a. Create a db user 'evm94'
 - As postgres,

```
$PG_HOME/bin/createuser --superuser --createdb --encrypted --pwprompt evm94
```

- b. Create a directory for db tablespace 'evm94'
 - As postgres,

```
cd $PG_HOME/data/pg_tblspc
mkdir evm94
```

- c. From pgAdmin3 GUI, create a 'evm94' tablespace

```
Name=evm94
Owner=evm94
Location=$PG_HOME/data/pg_tblspc/evm94
Tablespace=evm94
```

- d. Create db 'evm94'
 - As postgres,

```
$PG_HOME/bin/createdb evm94 --owner=evm94 --tablespace=evm94
```

- Execute the Trueview TNP database update script to add required tables to the created database table:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>/database_scripts
$ ./databaseUpdate.sh
```

You will be prompted to enter database parameters. The following is the console output of the script:

```
Do you want to continue? (y/n)
y
Please enter the database type for this installation (oracle, postgres):
postgres
Please enter database user name:
<dbusernameTNP>
Please enter database user password:
<dbpasswordTNP>
Please enter the database server host name or ip:
<localhost> or <IPaddr> or <dbhostname>
Please enter the database server port(5432):
<dbserverport>
Please enter the tablespace name for data:
<dbtablespaceTNP>
Please enter the tablespace name for indexes:
<dbtablespaceTNP>
Please enter the LDAP authentication scheme for this installation (openldap,
activedirectory): openldap
```

The script will finish with successful messages such as the following:

```
Updating Change Set Checksums ...
Migration successful
Updating Application Schema ...
Migration successful
Updating Application Privileges ...
Migration successful
```



NOTE: Since all required Trueview database tables are being created by these scripts, the total running time might be close to 2-4 minutes. Confirm that three “Migration successful” messages are returned from the update script as it runs. If any one step is unsuccessful, the other will not run – contact HEWLETT PACKARD ENTERPRISE Support for assistance with any error messages on installation/configuration.

- If the environment is making use of Event Management , then execute the Trueview EVM database update script to add required tables to the created database table:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>/database_scripts
$ ./evm-databaseUpdate.sh
```

You will be prompted to enter database parameters. The following is the console output of the script:

```
Do you want to continue? (y/n)
y
Please enter the database type for this installation (oracle, postgres):
postgres
Please enter database user name:
<dbusernameEVM>
Please enter database user password:
```

```

<dbpasswordEVM>
Please enter the database server host name or ip:
<localhost> or <IPaddr> or <dbhostname>
Please enter the database server port(5432):
<dbserverport>
Please enter the tablespace name for data:
<dbtablespaceEVM>
Please enter the tablespace name for indexes:
<dbtablespaceEVM>
Please enter the LDAP authentication scheme for this installation (openldap,
activedirectory): openldap

```

The script will finish with successful messages such as the following:

```

Updating Application Schema ...
Migration successful

```



NOTE: Since all required Trueview database tables are being created by these scripts, the total running time might be close to 2-4 minutes. Confirm that one "Migration successful" message is returned from the update script as it runs. If any one step is unsuccessful, the other will not run – contact HEWLETT PACKARD ENTERPRISE Support for assistance with any error messages on installation/configuration.

2.3.4 Setup TNP logging

1. Custom log4j.xml file

```

mkdir $TNP_HOME/custom
Place a <log4j>.xml into custom/

```



NOTE: \$TNP_HOME should be pointing to <webserver_homeDir>/tnp-config/<tnpwebobjectname>

2. Modify tnp.war to exclude the JBoss log4j module

```

mkdir $TNP_HOME/distrib/tnp.war.modifyForLog4j
cd $TNP_HOME/distrib/tnp.war.modifyForLog4j
jar xvf ../tnp.war
cd WEB-INF

```

3. Modify jboss-deployment-structure.xml by adding this tag inside <deployment>

```

<exclusions>
  <module name='org.apache.log4j' />
</exclusions>
cd ..
jar cvf ../tnp.war.log4j *
cd ..
mv tnp.war tnp.war.ori
ln -s tnp.war.log4j tnp.war

```

2.3.5 Trueview Database Application Configuration

To integrate the Trueview configuration directory with the new database, the Trueview configuration table needs to be updated. It is recommended that this is done prior to Trueview being started up. Using any valid SQL tool, connect to the database <dbtablespaceTNP> and execute the following:

```
update ensemble_config
set value = '<webserver_homeDir>/tnp-config/<tnpwebobjectname>'
where name like '%ApplicationConfigDirectory%' ;
commit;
```



NOTE: If no SQL tool (or SQL access) is available, this change can be made through the Trueview GUI once Trueview has been started for the first time. Once this change has been made, Trueview will require an immediate stop and restart to capture the change. Refer back to this step once Trueview has been started in the appropriate web server section of this guide.

With Trueview started by the appropriate web server mechanism, login to Trueview as a 'Trueview Administrator' user. After logging in, navigate to:

Configuration → System → Directories tab

And set the 'Application Config Directory' field (highlighted in yellow below) to the value:

<webserver_homeDir>/tnp-config/<tnpwebobjectname>

The screenshot displays the 'System Configuration: TNP_9011' interface. At the top, there are navigation tabs for Dashboard, Video Delivery, DCM, Requests, Orchestration, Services, Documents, Views, Digital Bridge, Reports, CMDB, System, and Admin. The 'System' tab is selected, and a dropdown menu is open, showing options like Security, Schedules, Process Templates, Interface, CMDB, Notifications, EVM, System, XML Configuration, Workbench, Customization, Product Catalog, and SR Workflows. The 'System' option is highlighted. Below the navigation, the 'System Configuration: TNP_9011' page is shown. It has a header with 'Server Id: TNP_9011', 'Database Username: TNP_LEL_9011_VSH', and 'Database Connection: jdbc:oracle:thin:@perses.tieroneoss.com:1521:orcl'. Below this, there are several tabs: Parameters, Directories, Event Handlers, User Interface, Authentication, Rules, Scheduling, Reconciliation, Workflow Engine, Languages, Syslog, and Service Management. The 'Directories' tab is active, showing a list of configuration fields: Application Config Directory(*) (highlighted in yellow with the value /home/weblogic/TNP_9011/), Adapter Home Directory (/home/weblogic/TNP_9011/adapters/), Connectors Home Directory (/home/weblogic/TNP_9011/connectors/), Custom Configuration Home Directory (/home/weblogic/TNP_9011/custom/), Provisioner Home Directory (/home/weblogic/TNP_9011/prov-configuration/), SVG Home Directory (/home/weblogic/TNP_9011/svg/), and Images Home Directory (/home/weblogic/TNP_9011/images/). A note at the bottom left says '(*) - Requires System Restart'.

Figure 1 System Configuration

Click on the Save icon in the upper right corner and then stop and restart Trueview through the appropriate web server mechanism.

2.4 Configuring Trueview on a Web Server

2.4.1 WebLogic Configuration and Startup

The WebLogic configuration script will create the new domain, admin server and Trueview managed server, install Trueview web applications and create the default Trueview admin user account. The script requires the use of the admin server, which in most cases is already setup and running. If not, then the script will take care of starting it up.

Prerequisites

- WebLogic 12.1.2 is installed
 - WebLogic user account is created
 - The WebLogic URL, which includes the port of the administration server (if already configured)
 - The name of the WebLogic admin server (if already configured)
 - The WebLogic admin user and password (if already created)
 - Database access detail
 - Node Manager is running.
1. Log on as **weblogic** user on Unix-based server and confirm that WL_HOME is set for the account. If it is not, then set it to the appropriate value through the appropriate commands for the shell being used.

```
$ env | grep WL_HOME
```



NOTE: \$WL_HOME may not be the same as <webserver_homeDir>, the home directory for the weblogic user.

2. Set WebLogic Environment Variables to run the WebLogic script:

```
$ cd $WL_HOME/server/bin
$ . ./setWLSEnv.sh
```



NOTE: There is a space in this command between the first and second periods.

3. Execute the 'Configure WebLogic' script:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>/bin
$ java weblogic.WLST configureWL.py
```

You will be prompted to enter WebLogic and Trueview configuration parameters. The values entered below are *suggested*. The following is the console output of the script:

```
Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Please Enter Admin ServerName: <adminservername>
Please enter Admin listen Port: <adminserverport>
Please Enter Managed Server Name: <tnpwebobjectname>
Please Enter Managed Server Listening Port: <tnpwebobjectport>
Please Enter the Database Server Host Name or IP: <localhost> or <IPaddr> or
<dbhostname>
Please Enter The Database Server Port: <dbserverport>
Please Enter the Oracle Database SID: <SID>
Please Enter TNP DB User: <dbusernameTNP>
Please Enter TNP DB Pwd: <dbpasswordTNP>
Please Enter WL Admin User Name: <webadminusername>
Please Enter WL Admin Password: <webadminpassword>
Please Enter WL Domain Path: $WL_HOME/user_projects/domains
Please Enter Domain Name: <tnpwldomainname>
Please Enter Trueview admin User Password: <tnpadminpassword>
```

```
Please Enter Trueview App Home Directory:
<webserver_homeDir>/tnp-config/<tnpwebobjectname>
Please Enter Trueview License File Name: <tnplicense>
```

The end of the console output will look like the following. If it does not, then review the text to troubleshoot the issue.

```
Disconnected from weblogic server: <adminservername>
Disconnected from weblogic server:
Exiting WebLogic Scripting Tool.
```

4. Start the WebLogic admin server.

```
$ cd $WL_HOME/user_projects/domains/<tnpwlomainname>
```



NOTE: Depending on how WebLogic is installed, this directory may differ slightly from the above.

```
$ . ./startWeblogic.sh &
```



NOTE: There is a space in this command between the first and second periods.

5. After the admin server has been started, log into the WebLogic Administration Console via a browser.

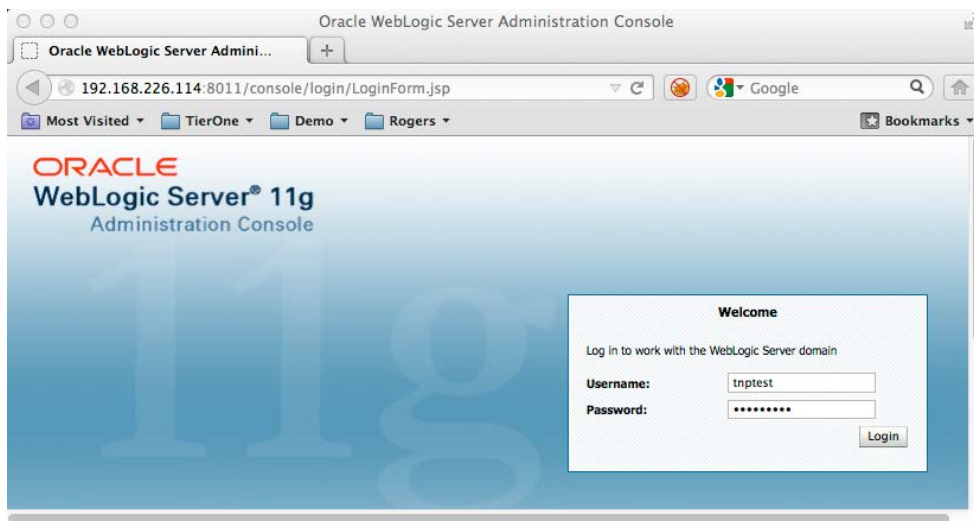


Figure 2 WebLogic Administration Console

Use <webadminusername> and <webadminpassword> as specified in a previous step.

6. Confirm the all necessary managed servers are present by selecting **Environment** – **Servers** under the Domain Structure frame.

There will be two servers, with the second server existing only if Event Management is being used.

- <tnpwebobjectname> on port <tnpwebobjectport>
- <evmwebobjectname> on port <evmwebobjectport>

If one or the other do not exist, then it will be necessary to manually create them through WebLogic by selecting New while in the **Environment – Servers** tab. All default values other than name and port can be specified.

7. Confirm that all necessary data sources are present by selecting Services – Data Sources under the Domain Structure frame.

There will be two data sources, with the second data source existing only if Event Management is being used.

- <tnpdatasource> for JNDI name TNP.database
- <evmdatasource> for JNDI name evm/datasource

If one or the other do not exist, then it will be necessary to manually create them through WebLogic by selecting New while in the **Services – Data Sources** tab. The (Oracle) values for the data sources are outlined as follows, adjust accordingly for Postgresql:

```
<tnpdatasource>
Name = DS_<tnpwebobjectname>
Target = <tnpwebobjectname>
JNDI Name = TNP.database
Database Type = Oracle
Database Driver = Oracle's Driver(Thin)for Service connections Ver9.0.1 and later
Database Name = <SID>
Host Name = <dbhostname>
Port = <dbserverport>
Database User = <dbusernameTNP>
Database User Password = <dbpasswordTNP>
MaxCapacity = 30
InitialCapacity = <MaxCapacity>
Shrink Frequency = 0
Test Connections On Reserve = <true>
Test Frequency = 0
```

```
<evmdatasource>
Name = DS_<evmwebobjectname>
Target = <evmwebobjectname>
JNDI Name = evm/datasource
Database Type = Oracle
Database Driver = Oracle's Driver(Thin XA)for Service connections Ver9.0.1 and
later
Database Name = <SID>
Host Name = <dbhostname>
Port = <dbserverport>
Database User = <dbusernameEVM>
Database User Password = <dbpasswordEVM>
MaxCapacity = 30
InitialCapacity = <MaxCapacity>
Shrink Frequency = 0
Test Connections On Reserve = <true>
Test Frequency = 0
XA Transaction Timeout = 600
```

8. Under 'Domain Structure,' select Services → JTA. Fill in the following field as specified and hit save:

Timeout Seconds = 600

9. Confirm that all necessary deployments are present by selecting Deployments under the Domain Structure frame.

The list should include the following objects:

- tnp
- tnp-ws
- tnp-report
- tnp-custom (if applicable)
- evm (if applicable)

If one of the deployments do not exist, then it will be necessary to manually create them through WebLogic by selecting New while in the **Deployments** tab. The deployments will all be created as applications with the non-default values outlined as follows:

```
Name = tnp
File = <webserver_homeDir>/tnp-config/<tnpwebobjectname>/release/distrib/tnp.war
Target = <tnpwebobjectname>
```

```
Name = tnp-ws
File = <webserver_homeDir>/tnp-config/<tnpwebobjectname>/release/distrib/tnp-
ws.war
Target = <tnpwebobjectname>
```

```
Name = tnp-report
File = <webserver_homeDir>/tnp-config/<tnpwebobjectname>/release/distrib/tnp-
report.war
Target = <tnpwebobjectname>
```

```
Name = tnp-custom
File = <webserver_homeDir>/tnp-config/<tnpwebobjectname>/custom/tnp-custom.war
Target = <tnpwebobjectname>
```

```
Name = evm
File = <webserver_homeDir>/tnp-
config/<tnpwebobjectname>/release/distrib/EventManagerAPP.ear
Target = <evmwebobjectname>
```

10. Under 'Domain Structure,' select **Environment** → **Servers** → <tnpwebobjectname> → **Server Start**. Fill in the two following fields as specified:

```
Class Path = $WL_HOME/server/lib/weblogic.jar:<webserver_homeDir>/
tnp-config/<tnpwebobjectname>/lib/
bcprov-jdk15-145.jar:<webserver_homeDir>/
tnp-config/<tnpwebobjectname>/lib/truecontrol-client.jar
```

If Weblogic is connecting to Postgresql add the following jar file to the above class path

```
:<webserver_homeDir>/
tnp-config/<tnpwebobjectname>/lib/postgresql-9.1-901.jdbc4.jar
```



NOTE: Each directory should be separated with a colon and there should be no spaces before or after the colon, or in any of the directory names.

```
Arguments = -Dserverid=<tnpwebobjectname>
-DlicenseFile=<webserver_homeDir>/
tnp-config/<tnpwebobjectname>/license/<tnplicense>
-Dreportconfigdir=<webserver_homeDir>/
tnp-config/<TNP_ServerPort>/reports
-Djava.awt.headless=true
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```



NOTE: Each value begins with a hyphen followed by no space and then ends with a space.

For clarity, the following diagram depicts an example of how to fill in the fields. Please note that the actual field content could differ from this example.

The screenshot shows the 'Server Start' configuration page in the WebLogic console. The 'Web Services' section is active, and the 'Save' button is visible. The configuration fields are filled with example values:

- Java Home:** /home/weblogic/Oracle/Middleware/wlserver_10.3.4/server
- Java Vendor:** BEA
- BEA Home:** /home/weblogic
- Root Directory:** /home/weblogic
- Class Path:** /home/weblogic/Oracle/Middleware/wlserver_10.3.4/server:/lib/weblogic.jar:/home/weblogic/tnp-config/TNP_8011/lib/beprov-jdk15-145.jar:/home/weblogic/tnp-config/TNP_8011/lib/truecontrol-client.jar
- Arguments:** -Dserverid=TNP_8011 -DlicenseFile=/home/weblogic/tnp-config/TNP_8011/license/tnp.lic -Dreportconfigdir=/home/weblogic/tnp-config/TNP_8011/reports -Xms256m -Xmx512m -XX:MaxPermSize=256m -Djava.awt.headless=true -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false

Figure 3 WebLogic configuration example

If the environment is making use of Event Management, there are no changes required for <evmwebobjectname> .

- In order to start the WebLogic Managed Servers for Trueview via script, the Class Path and Arguments must be added to the setDomainEnv.sh script and USER_MEM_ARGS to the startWebLogic.sh script.

```
$ cd $WL_HOME/user_projects/domains/<tnpwlomainname>/bin
```

Open the setDomainEnv.sh file for editing.

- Copy the Class Path parameters from the WebLogic Console to the end of setDomainEnv.sh file. Add the CLASSPATH environment variable syntax around the text pasted. It will look like the text below:

```
CLASSPATH="${CLASSPATH}:${WL_HOME}/server/lib/weblogic.jar:
<webserver_homeDir>/tnp-config/<tnpwebobjectname>/lib/
bcprov-jdk15-151.jar"
export CLASSPATH
```



NOTE: The value after the equal sign begins and ends with a double quote and each directory inside the double quotes are separated with a colon. There should be no spaces inside the double quotes.

13. Copy the Arguments parameters from the WebLogic Console to the end of `setDomainEnv.sh` file. Add the `JAVA_OPTIONS` environment variable syntax around the text pasted. It will look like the text below:

```
JAVA_OPTIONS="${JAVA_OPTIONS} -Dserverid=<tnpwebobjectname>
-DlicenseFile=<webserver_homeDir>/
tnp-config/<tnpwebobjectname>/license/<tnplicense>
-Dreportconfigdir=<webserver_homeDir>/
tnp-config/<TNP_ServerPort>/reports
-Djava.awt.headless=true
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false"
export JAVA_OPTIONS
```



NOTE: The value after the equal sign begins and ends with a double quote. Also, each value inside the double quotes begins with a hyphen followed by no space and then ends with a space.

14. Save the `setDomainEnv.sh` file.
15. Open the `startWebLogic.sh` file for editing.
16. Alter the file by adding a case statement to ensure the correct values are defined for the appropriate managed server. The section will now look something like this:

```
case $SERVER_NAME in
  <tnpwebobjectname>)
    USER_MEM_ARGS="-XX:+UseParNewGC -XX:ParallelGCThreads=2 -
XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=40 -XX:SurvivorRatio=2 -
Xms1024m -Xmx1024m -Xmn128m -XX:PermSize=256m -XX:MaxPermSize=512m"
    export USER_MEM_ARGS
    ;;
  <evmwebobjectname>)
    USER_MEM_ARGS="-XX:+UseParNewGC -XX:ParallelGCThreads=2 -
XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=40 -XX:SurvivorRatio=2 -
Xms1024m -Xmx1024m -Xmn128m -XX:PermSize=256m -XX:MaxPermSize=512m"
    export USER_MEM_ARGS
    ;;
  *)
    USER_MEM_ARGS="-XX:+UseParNewGC -XX:ParallelGCThreads=2 -
XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=40 -XX:SurvivorRatio=2 -
Xms512m -Xmx512m -Xmn128m -XX:PermSize=256m -XX:MaxPermSize=512m"
    export USER_MEM_ARGS
    ;;
esac
```

17. In order to allow WebLogic to perform LDAP-based authentication for Trueview, the default security realm within the WebLogic server must be modified. In general, the default realm is designated as "myrealm". This realm should have a "DefaultAuthenticator" which is used to authenticate basic system users (such as Weblogic, OracleSystemUser, etc...). The following steps assume a default realm called "myrealm" which has a "DefaultAuthenticator" associated to it. Modify the security realm "myrealm" as follows:
 - a. Security Realms Node
 - i. Navigate to the "Security Realms" node from the tree control and select "myrealm."
 - b. Providers → Authentication Tab

- i. Navigate to the “Providers → Authentication” Tab for the “myrealm” Security Realm.
 - ii. Select the “DefaultAuthenticator.”
- c. Configuration → Common Tab
 - i. From the “Configuration → Common” Tab, set the “Control Flag” parameter to: SUFFICIENT.
 - ii. Save the changes.
- d. Providers → Authentication Tab
 - i. Navigate to the “Providers → Authentication” Tab for the “myrealm” Security Realm.
 - ii. Create a new Authentication Provider.
 - iii. Set the “Name” parameter to a suitable value which will identify this Provider (e.g. LDAP-C).
 - iv. Set the “Type” parameter to a value of “LDAPAuthenticator.”
 - v. Reorder the Authentication Providers as follows: LDAP-C, Default Authenticator.
 - vi. Select the newly created “LDAP-C” Authentication Provider.
- e. Configuration → Common Tab
 - i. Navigate to the “Configuration → Common” Tab for the newly created “LDAP-C” Authentication Provider.
 - ii. Set the “Control Flag” parameter to a value of “SUFFICIENT.”
 - iii. Save the changes.
- f. Configuration → Provider Specific Tab (Connection)
 - i. Navigate to the “Configuration → Provider Specific” Tab for the newly created “LDAP-C” Authorization Provider.
 - ii. Locate the “Connection” section.
 - iii. Set the “Host” = <ldaphostname>
 - iv. Set the “Port” = <ldaphostport>
 - v. Set the “Principal” = <ldapprincipal>
 - vi. Set the “Credential” = <ldapcredential>
 - vii. Set the “Confirm Credential” = <ldapcredential>
 - viii. Select the “SSEnabled” check box
- g. Configuration → Provider Specific Tab (Users)
 - i. Locate the “Users” section
 - ii. Set the “User Base DN” = <ldapuserbasedn>
 - iii. Set the “User From Name Filter” = (&(sAMAccountName=%u)(objectclass=user))
 - iv. Set the “User Search Scope” = subtree
 - v. Set the “User Name Attribute” = sAMAccountName
 - vi. Set the “User Object Class” = user
- h. Configuration → Provider Specific Tab (Groups)
 - i. Locate the “Groups” section
 - ii. Set the “Group Base DN” = <ldapgroupbasedn>
 - iii. Set the “Group From Name Filter” = (&(cn=%g)(objectclass=group))
 - iv. Set the “Group Search Scope” = subtree
 - v. Set the “Group Membership Searching” = limited
 - vi. Set the “Max Group Membership Search Level” = 0
- i. Configuration → Provider Specific Tab (Static Groups)
 - i. Locate the “Static Groups” section
 - ii. Set the “Static Group Name Attribute” = cn
 - iii. Set the “Static Group Object Class” = group
 - iv. Set the “Static Member DN Attribute” = member
 - v. Set the “Static Group DN from Member DN Filter” = (&(member=%M)(objectclass=group))
- j. Configuration → Provider Specific Tab (General)
 - i. Locate the “General” section.
 - ii. Set the “Connection Pool Size” = 6
 - iii. Set the “Connect Timeout” = 3
 - iv. Set the “Connection Retry Limit” = 1
 - v. Set the “Parallel Connect Delay” = 0
 - vi. Set the “Results Time Limit” = 0
 - vii. Select the “Follow Referrals” check box
 - viii. Select the “Cache Enabled” check box
 - ix. Set the “Cache Size” = 32
 - x. Set the “Cache TTL” = 60

- xi. Set the “GUID Attribute” = `objectguid`
- xii. Save the changes
- xiii. Restart the WebLogic server.
- k. Roles and Policies Tab
 - i. Navigate to the “Roles and Policies” Tab for the “myrealm” Security Realm.
 - ii. Within “Global Roles → Roles” create a new Role.
 - iii. Set the “Name” parameter to the Role value defined in the “weblogic.xml” file (e.g. TNP).
 - iv. Set the “Provider Name” parameter to: `XACMLRoleMapper`.
 - v. Select the new Role (e.g. TNP).
 - vi. Set the “Role Conditions” to ‘Allow access to everyone.’
 - vii. A need may arise in which it may be desirable to allow non LDAP users which belong to other Authentication Providers (e.g. DefaultAuthenticator) access to Trueview. These users may be given access by simply adding them as subsequent Role Conditions to the TNP Role.
 - viii. Save the changes.
 - ix. Apply all changes and restart the entire WebLogic server.
 - x. Users and Groups
 - xi. Navigate to the “Users and Groups” Tab for the “myrealm” Security Realm.
 - xii. The following users should exist. If not, they need to be added.
- 18. If the environment is making use of Event Management, then it is necessary to confirm that a necessary JMS server is present by selecting Services – Messaging – JMS Servers under the Domain Structure frame.

There will be one server.

- `JMS_<evmwebobjectname>`

If the server does not exist, then it will be necessary to manually create them through a script provided. Execution of the script is as follows:

```
$ WL_HOME=<weblogic_homeDir>; export WL_HOME
$ JAVA_HOME=<java_homeDir>; export JAVA_HOME
$ cd <webserver_homeDir>/tnp-config/<TNP_ServerPort>/bin
$ ./evmJMSConfig.sh <evmwebobjectname> <webadminusername> <webadminpassword>
t3://<adminservername>:<adminserverport>
```

- 19. If the environment is making use of Event Management, then it is necessary to configure EVM prior to startup.

First generate an encrypted pw for the main administration user of TNP, <tnpadminuser> though the tool `syncApp.sh`.

```
$ TNP_HOME=<Trueview_appDir>; export TNP_HOME
$ JAVA_HOME=<java_homeDir>; export JAVA_HOME
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>
$ bin/syncApp.sh -e
```

When requested, enter in the valid password for <tnpadminuser> and the script will return an encrypted password.

Next create a custom `coreConfig.xml` file to contain this encrypted password and some additional values:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>
$ cp -p release/samples/coreConfig.xml custom/
$ cd custom
```

Open the file `coreConfig.xml` for editing and make the following adjustments:

```
RTU_POLL_INTERVAL = 5
RTU_LIMIT_PER_INTERVAL = 10
TNP_INVENTORY_URL = http://<adminservername>:<tnpwebobjectport>/tnp-ws
```



```
TNP_INVENTORY_USERNAME = <tnpadminuser>
TNP_INVENTORY_PASSWORD = <tnpadminuser_encryptedpasswd>
TNP_COM_ROUTING_ENABLED = false
```

Finally, populate the EVM database with the configuration values:

```
$ ../bin/evmConfig.sh
http://<adminservername>:<evmwebobjectport>/evmapi/ConfigurationSubmit/ConfigurationSubmitHttp coreConfig.xml
```

If the command fails with 'Connection refused', then the parameters within in coreConfig.xml will need to be entered manually into the EVM database's EVMCONFIG table via sql.

20. Start the Trueview Managed Server from either the WebLogic Console or from the startManagedWeblogic.sh script.
21. To log into Trueview, open a browser and enter one of the following:

```
http://<hostname>:<tnpwebobjectport>/tnp
```



NOTE: <hostname> can be replaced with <ipaddr>.

22. If the environment is making use of Event Management, it is necessary to sync the content of the EVM database with that of TNP. This is accomplished via the following:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>
$ mkdir custom
$ cp -p release/samples/SyncApp.properties custom/
$ cd custom
```

Open the file `SyncApp.properties` for editing and make the following adjustments:

```
INVENTORY_URL = http://<adminservername>:<tnpwebobjectport>/tnp-ws/alarmDependency
TNP_INVENTORY_USERNAME = <tnpadminuser>
TNP_INVENTORY_PASSWORD = <tnpadminuser_encryptedpasswd>
EVM_URL = http://<adminservername>:<evmwebobjectport>/
evmapi/InventorySync/InventorySyncHttp
```

Finally execute the script with any or all of the following parameters. In some instances, it is necessary to first run the network element (n) options first, followed by the facilities (f) and circuits (c).

```
$ ../bin/syncApp.sh -nd -na -fd -fa -cd -ca
```

Although the script appears to finish, the execution is still ongoing and can take some time to finish. To confirm execution has been completed is to count the number of rows in the EVM database table UPDATEQUEUE.

As the typical that the user environment may not have all the correct settings, it is recommend to create a wrapper script for the SyncApp.sh script.

Open the file `SyncApp.sh` for editing and add the following lines (adjusting as appropriate)

```
#!/bin/sh
# The JAVA_HOME and TNP_HOME variables need to be set
# either in the environment or by uncommenting these entries
TNP_HOME=<netporal_appDir>
export TNP_HOME
JAVA_HOME=<java_homeDir>
```

```
export JAVA_HOME
${TNP_HOME}/bin/syncApp.sh $@
```

2.4.2 Jboss Installation, Configuration, and Startup



NOTE: Event Management is not supported in Jboss.

2.4.2.1 JBoss Installation

1. User Account Setup
 - i) As root, create user 'jboss'

```
useradd -c "JBoss Owner" -d /home/jboss -g users -m -s /bin/bash jboss
passwd jboss
```

- ii) As root, install the 'unzip' utility as needed

```
yum install unzip
```

- iii) As root, create 'jboss' app directory

```
mkdir /opt/jboss
chown jboss:users /opt/jboss
```

2. JBoss Zip File
 - i) Download Red Hat JBoss Enterprise Application Platform 6.2.0.GA zip file from JBossDeveloper (<https://www.jboss.org/products/eap/download/>) to host:/opt/TAR/
 - ii) As jboss, extract the zip file under /opt/jboss/

```
cd /opt/jboss
unzip /opt/TAR/jboss-eap-6.2.0.zip
```

The contents are extracted to /opt/jboss/jboss-eap-6.2 (aka *JBOSS_HOME*)

3. JBoss Environment Variables

Set the following environment variables in ~jboss/.bash_profile

```
JAVA_HOME=/usr/java/jdk1.7.0_79/jre
JBOSS_HOME=/opt/jboss-eap-6.2
EAP_HOME=$JBOSS_HOME
TNP_HOME=/opt/jboss/tnp-config/8180
JBOSS_CLASSPATH=$TNP_HOME/lib/bcprov-jdk15on-151.jar:$TNP_HOME/lib/xbean.jar:$TNP_HOME/lib/jsoup-1.7.3.jar:$JAVA_HOME/lib/rt.jar
PATH=$PATH: $HOME/bin:$JAVA_HOME/bin:$TNP_HOME/bin:$JBOSS_HOME/bin
export JAVA_HOME JBOSS_HOME EAP_HOME TNP_HOME JBOSS_CLASSPATH PATH
```

4. JBoss Configuration for Standalone Mode

1. Edit configuration file `$JBOSS_HOME/bin/init.d/jboss-as.conf`
 - `JBOSS_USER=jboss`
 - `JBOSS_HOME=/opt/jboss/jboss-eap-6.2`
2. Link `jboss-as.conf` into system directory
 - i. create directory `/etc/jboss-as/` owned by `jboss:users`

```
cd /etc/jboss-as
ln -s $JBOSS_HOME/bin/init.d/jboss-as.conf .
```

3. Edit configuration file `$JBOSS_HOME/bin/init.d/jboss-as-standalone.sh`

```
JBOSS_CONFIG=standalone-full.xml
```

4. Link `jboss-as-standalone.sh` into system directory

```
ln -s $JBOSS_HOME/bin/init.d/jboss-as-standalone.sh /etc/init.d/jboss
```

5. Add jboss startup script as a service

```
[as root]
chkconfig --add jboss
```

6. Make JBoss auto-start on server bootup

```
[as root]
chkconfig jboss on
```

7. Create admin user 'jbadmin' for JBoss

```
$JBOSS_HOME/bin/add-user.sh
```

5. Setup JBoss datasource to PostgreSQL
 - i) JDBC driver

```
mkdir -p $JBOSS_HOME/modules/jdbc/postgresql/main
cd $JBOSS_HOME/modules/jdbc/postgresql/main
```

- Download a PostgreSQL JDBC driver (`postgresql-9.4.1207.jre7.jar`) into directory
- Create `module.xml` with content

```
<?xml version='1.0' encoding='UTF-8'?>
<module xmlns="urn:jboss:module:1.0" name="jdbc.postgresql">
  <resources>
    <resource-root path="postgresql-9.4.1207.jre7.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

- ii) Setup `module.xml` and library links in `JBOSS_HOME/modules/tnplib/lib/jaxws/main/`

```
mkdir -p $JBOSS_HOME/modules/tnplib/lib/jaxws/main
cd $JBOSS_HOME/modules/tnplib/lib/jaxws/main
```

➤ Create module.xml with content

```
<module xmlns="urn:jboss:module:jaxws:1.1" name="tnplib.lib.jaxws">
  <resources>
    <resource-root path="gmbal-api-only.jar"/>
    <resource-root path="management-api.jar"/>
    <resource-root path="policy.jar"/>
    <resource-root path="stax-ex.jar"/>
    <resource-root path="streambuffer.jar"/>
    <resource-root path="jaxb-impl.jar"/>
    <resource-root path="jaxb-xjc.jar"/>
    <resource-root path="jaxws-rt.jar"/>
    <resource-root path="bcprov-jdk15on-151.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.xml.bind.api"/>
    <module name="javax.xml.stream.api"/>
    <module name="javax.servlet.api"/>
    <module name="javax.xml.ws.api"/>
    <module name="javax.jms.api"/>
    <module name="javax.jws.api"/>
    <module name="javax.annotation.api"/>
  </dependencies>
</module>
```

```
ln -s $JBOSS_HOME/standalone/lib/ext/gmbal-api-only.jar .
[*needed to download jar]
ln -s $JBOSS_HOME/standalone/lib/ext/management-api.jar .
[*needed to download jar]
ln -s $JBOSS_HOME/standalone/lib/ext/policy.jar .
[*needed to download jar]
ln -s $TNP_HOME/lib/jaxws-2.1.7/stax-ex.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/streambuffer.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/jaxb-impl.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/jaxb-xjc.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/jaxws-rt.jar .
ln -s $TNP_HOME/lib/bcprov-jdk15on-151.jar .
```

iii) Setup module.xml and library links in JBOSS_HOME/modules/tnplib/lib/xbean/main/

```
mkdir -p $JBOSS_HOME/modules/tnplib/lib/xbean/main
cd $JBOSS_HOME/modules/tnplib/lib/xbean/main
```

➤ Create module.xml with content

```
<module xmlns="urn:jboss:module:xbean:1.1" name="tnplib.lib.xbean">
  <resources>
    <resource-root path="xbean.jar"/>
    <resource-root path="jsoup-1.7.3.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
```

```

</dependencies>
</module>
ln -s $TNP_HOME/lib/xbean.jar .
ln -s $TNP_HOME/lib/jsoup-1.7.3.jar .

```

- iv) Setup `module.xml` in `JBOSS_HOME/modules/system/layers/base/sun/jdk/main/`

```
cd $JBOSS_HOME/modules/system/layers/base/sun/jdk/main
```

- insert these 2 lines inside the `<paths>` tag in `module.xml`

```

<path name="sun/net/www/protocol/http"/>
<path name="sun/net/www/protocol/https"/>

```

6. Setup JBoss deployment files for Trueview

```
cd $JBOSS_HOME/standalone/deployments
ln -s $TNP_HOME/distrib/* .
```

7. Update `$JBOSS_HOME/standalone/configuration/standalone-full.xml`

- i) Add this `<datasource>` tag for TNP within the `<datasources>` tag. Modify these values as necessary.

```

<datasource jndi-name="java:/TNP.database" pool-name="TNP.database" enabled="true"
use-java-context="true">
  <connection-url>jdbc:postgresql://192.168.229.92:5432/tnp94</connection-url>
  <driver-class>org.postgresql.Driver</driver-class>
  <driver>postgresql</driver>
  <pool>
    <min-pool-size>2</min-pool-size>
    <max-pool-size>20</max-pool-size>
    <prefill>true</prefill>
  </pool>
  <security>
    <user-name>tnp94</user-name>
    <password>tnp94</password>
  </security>
  <validation>
    <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
    <validate-on-match>>false</validate-on-match>
    <background-validation>>false</background-validation>
    <use-fast-fail>>false</use-fast-fail>
  </validation>
</datasource>

```

- ii) Add this `<datasource>` tag for EVM within the `<datasources>` tag. Modify these **values** as necessary.

```

<datasource jndi-name="java:/evm/datasource" pool-name="EVM.database"
enabled="true" use-java-context="true">
  <connection-url>jdbc:postgresql://192.168.229.92:5432/tnp94</connection-url>
  <driver-class>org.postgresql.Driver</driver-class>
  <driver>postgresql</driver>
  <pool>
    <min-pool-size>2</min-pool-size>
    <max-pool-size>20</max-pool-size>
    <prefill>true</prefill>
  </pool>
  <security>

```

```

    <user-name>tnp94</user-name>
    <password>tnp94</password>
  </security>
  <validation>
    <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
    <validate-on-match>>false</validate-on-match>
    <background-validation>>false</background-validation>
    <use-fast-fail>>false</use-fast-fail>
  </validation>
</datasource>

```

- iii) Add this <driver> tag within the <drivers> tag

```

<driver name="postgresql" module="jdbc.postgresql">
  <xa-datasource-class>jdbc.postgresql.xa.PGXADatasource</xa-datasource-class>
</driver>

```

- iv) Add this <global-modules> tag after the <subsystem xmlns="urn:jboss:domain:ee:1.1"> tag

```

<global-modules>
  <module name="tnplib.lib.xbean" slot="main"/>
  <module name="tnplib.lib.jaxws" slot="main"/>
</global-modules>

```

- v) Add this <security-domain> tag within the <security-domains> tag. Modify these **values** as necessary.

```

<security-domain name="TNPLDAP" cache-type="default">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.LdapExtLoginModule"
flag="required">
      <module-option name="password-stacking" value="useFirstPass"/>
      <module-option name="java.naming.factory.initial"
value="com.sun.jndi.ldap.LdapCtxFactory"/>
      <module-option name="java.naming.provider.url"
value="ldap://192.168.223.240:389"/>
      <module-option name="java.naming.security.authentication"
value="simple"/>
      <module-option name="bindDN" value="cn=root,dc=tieroneoss,dc=com"/>
      <module-option name="bindCredential" value="secret"/>
      <module-option name="baseCtxDN"
value="ou=LDAPusers,dc=tieroneoss,dc=com"/>
      <module-option name="baseFilter" value="(cn={0})"/>
      <module-option name="rolesCtxDN"
value="ou=LDAPgroups,dc=tieroneoss,dc=com"/>
      <module-option name="defaultRole" value="TNP"/>
      <module-option name="roleFilter" value="(member={1})"/>
      <module-option name="roleAttributeIsDN" value="true"/>
      <module-option name="roleNameAttributeID" value="cn"/>
    </login-module>
  </authentication>
</security-domain>

```

- vi) Set port value to 8180 for this <socket-binding> tag

```

<socket-binding name="http" port="8180"/>

```

8. Insert the following entries into \$JBOSS_HOME/bin/standalone.conf Provide these **values** appropriately.

```

JAVA_HOME="/usr/java/jdk1.7.0_79/jre"

JAVA_OPTS="-D[Server:server-one] -XX:PermSize=256m -XX:MaxPermSize=256m -Xms1303m
-Xmx1303m -server -Djboss.bind.address=JBoss svr IP -
Dsun.rmi.dgc.client.gcInterval=3600000 -Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true -
Djboss.modules.system.pkgs=org.jboss.byteman -Djava.net.preferIPv4Stack=true -
Dserverid=JBoss svrName -Djboss.bind.address.management=JBoss svr IP -
Dsun.lang.ClassLoader.allowArraySyntax=true -
DlicenseFile=TNP_HOME/license/license.lic -
Dlog4j.configuration=file:TNP_HOME/path/to/log4j.xml"

```

2.4.2.2 JBoss (TNP) Start/Status/Stop

```

Start
sudo service jboss start
Status
sudo service jboss status
Stop
sudo service jboss stop

```

2.4.2.3 Trueview Installation

1. Create OS user Trueview

```

# useradd -c "Trueview owner" -d /home/Trueview -g staff -m -s /bin/bash Trueview
# mkdir /home/Trueview
# chown Trueview:staff /home/Trueview

```

2. Load New Build as user Trueview
 - a. copy TNP and ADP build files to repository <host>:<Trueview_homeDir>/app/.

```

cd <Trueview_homeDir>/app/
tar xvpfz <tnpBuild>.tar.gz
tar xvpfz <adpBuild>.tar.gz

```

3. Setup TNP application directory

As user jboss:

```

mkdir -p <Trueview_appDir>
cd <Trueview_appDir>
ln -s <Trueview_tarDir>/<tnpBuild> release
ln -s <Trueview_tarDir>/<adpBuild> release-adapters
ln -s release/api .
ln -s release/bin .
ln -s release/BUILD_ID .
ln -s release/database_scripts .
ln -s release/distrib .
ln -s release/lib .
ln -s release/prov-configuration .
ln -s release/svg .
ln -s release/utils .
cp -rp release-adapters/adapters .
cp -rp release-adapters/AlcatelSAMServiceDataConnector .
cp -rp release-adapters/connectors .
cp -rp release-adapters/NetSmartConnector .

```

```
cp -rp release/eventhandling .
cp -rp release/reports .
mkdir custom
mkdir license
mkdir logs
mkdir uploads
```

Place a <license>.lic into license/

Place a log4j.xml into custom/

Place a tnp-custom.war into custom/

4. Create TNP dbSchema
 - a. As Oracle admin, create a dbSchema for TNP
 - b. As user jboss, execute <Trueview_appDir>/database_scripts/databaseUpdate.sh

2.4.2.4 JBoss Configuration

JBoss can be configured for servers on 4 preset ports, where the "default" JBoss server port is 8080 and other servers use a port number that is "100" off of the "default" -- e.g. 8180, 8280, 8380. The TNP env will use port 8180.

1. Setup module.xml and libraries in JBOSS_HOME/modules/jdbc/oracle/main/

```
cd $JBOSS_HOME/modules/jdbc/oracle/main

vi module.xml

<?xml version='1.0' encoding='UTF-8'?>
<module xmlns="urn:jboss:module:1.1" name="jdbc.oracle">
  <resources>
    <resource-root path="ojdbc6-11.0.jar"/>
    <resource-root path="orai18n-11.0.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>

ln -s $TNP_HOME/lib/ojdbc6-11.0.jar .
ln -s $TNP_HOME/lib/orai18n-11.0.jar .
```

2. Setup module.xml and libraries in JBOSS_HOME/modules/tnplib/lib/jaxws/main/

```
cd $JBOSS_HOME/modules/tnplib/lib/jaxws/main
vi module.xml
<module xmlns="urn:jboss:module:jaxws:1.1" name="tnplib.lib.jaxws">
  <resources>
    <resource-root path="gmbal-api-only.jar"/>
    <resource-root path="management-api.jar"/>
    <resource-root path="policy.jar"/>
    <resource-root path="stax-ex.jar"/>
    <resource-root path="streambuffer.jar"/>
    <resource-root path="jaxb-impl.jar"/>
    <resource-root path="jaxb-xjc.jar"/>
    <resource-root path="jaxws-rt.jar"/>
  </resources>
  <dependencies>
```



```

    <module name="javax.api"/>
    <module name="javax.xml.bind.api"/>
  <module name="javax.xml.stream.api"/>
    <module name="javax.servlet.api"/>
    <module name="javax.xml.ws.api"/>
    <module name="javax.jms.api"/>
    <module name="javax.jws.api"/>
    <module name="javax.annotation.api"/>
  </dependencies>
</module>
ln -s $JBOSS_HOME/standalone/lib/ext/gmbal-api-only.jar .
ln -s $JBOSS_HOME/standalone/lib/ext/management-api.jar .
ln -s $JBOSS_HOME/standalone/lib/ext/policy.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/stax-ex.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/streambuffer.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/jaxb-impl.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/jaxb-xjc.jar .
ln -s $TNP_HOME/lib/jaxws-2.1.7/jaxws-rt.jar .

```

3. Setup module.xml and libraries in JBOSS_HOME/modules/tnplib/lib/xbean/main/

```

cd $JBOSS_HOME/modules/tnplib/lib/xbean/main
vi module.xml
<module xmlns="urn:jboss:module:xbean:1.1" name="tnplib.lib.xbean">
  <resources>
    <resource-root path="xbean.jar"/>
    <resource-root path="jsoup-1.7.3.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
ln -s $TNP_HOME/lib/xbean.jar .
ln -s $TNP_HOME/lib/jsoup-1.7.3.jar .

```

4. Setup the Trueview deployment war files

```

cd $JBOSS_HOME/standalone/deployments
ln -s $TNP_HOME/distrib/tnp.war .
ln -s $TNP_HOME/distrib/tnp-ws.war .
ln -s $TNP_HOME/distrib/tnp-report.war .
ln -s $TNP_HOME/custom/tnp-custom.war .

```

5. Modify \$JBOSS_HOME/standalone/configuration/standalone.xml
 - a. remove these elements

```

<extension module="org.jboss.as.configadmin"/>
<extension module="org.jboss.as.osgi"/>
<subsystem xmlns="urn:jboss:domain:configadmin:1.0"/>

```

- b. remove this subsystem element

```

<subsystem xmlns="urn:jboss:domain:osgi:1.2" activation="lazy">
  <properties>
    <!-- Specifies the beginning start level of the framework -->
    <property name="org.osgi.framework.startlevel.beginning">1</property>
  </properties>
  <capabilities>
    <!-- modules registered with the OSGi layer on startup -->

```

```

<capability name="javax.jws.api" />
<capability name="javax.persistence.api" />
<capability name="javax.servlet.api" />
<capability name="javax.transaction.api" />
<capability name="javax.ws.rs.api" />
<capability name="javax.xml.bind.api" />
<capability name="javax.xml.ws.api" />
<capability name="org.slf4j" />
<!-- bundles started in startlevel 1 -->
<capability name="org.apache.felix.log" startlevel="1" />
<capability name="org.jboss.osgi.logging" startlevel="1" />
<capability name="org.apache.felix.configadmin" startlevel="1" />
<capability name="org.jboss.as.osgi.configadmin" startlevel="1" />
<capability name="org.jboss.as.osgi.http" startlevel="1" />
<capability name="org.jboss.as.osgi.jpa" startlevel="1" />
</capabilities>
</subsystem>

```

- c. remove all security-domain elements in security-domains element
- d. insert this security-domain element in security-domains element

```

<security-domain name="TNPLDAP" cache-type="default">
<authentication>
<login-module code="org.jboss.security.auth.spi.LdapExtLoginModule"
flag="required">
<module-option name="password-stacking" value="useFirstPass" />
<module-option name="java.naming.factory.initial"
value="com.sun.jndi.ldap.LdapCtxFactory" />
<module-option name="java.naming.provider.url" value="ldap://<ldapSvrIP>:389" />
<module-option name="java.naming.security.authentication" value="simple" />
<module-option name="bindDN" value="cn=root,dc=<domainname>,dc=com" />
<module-option name="bindCredential" value="<ldapPW" />
<module-option name="baseCtxDN" value="ou=LDAPusers,dc=<domainname>,dc=com" />
<module-option name="baseFilter" value="(cn={0})" />
<module-option name="rolesCtxDN" value="ou=LDAPgroups,dc=<domainname>,dc=com" />
<module-option name="defaultRole" value="TNP" />
<module-option name="roleFilter" value="(member={1})" />
<module-option name="roleAttributeIsDN" value="true" />
<module-option name="roleNameAttributeID" value="cn" />
</login-module>
</authentication>
</security-domain>

```

- e. insert this datasource element inside the datasources element

```

<datasource jndi-name="java:/TNP.database" pool-name="TNP.database"
enabled="true" use-java-context="true">
<connection-url>jdbc:oracle:thin:@<dbSvrIP>:<dbSvrPort>:<sid></connection-url>
<driver-class>oracle.jdbc.OracleDriver</driver-class>
<driver>oracle</driver>
<security>
<user-name><dbUserId></user-name>
<password><dbUserPw></password>
</security>
<validation>
<validate-on-match>>false</validate-on-match>
<background-validation>>false</background-validation>
</validation>
<statement>
<share-prepared-statements>>false</share-prepared-statements>
</statement>

```

```
</datasource>
```

- f. insert this driver element inside the drivers element

```
<driver name="oracle" module="jdbc.oracle">
<xa-datasource-class>oracle.jdbc.driver.OracleDriver</xa-datasource-class>
</driver>
```

- g. insert this global-modules element before the spec-descriptor-property-replacement element

```
<global-modules>
<module name="tnplib.lib.xbean" slot="main"/>
<module name="tnplib.lib.jaxws" slot="main"/>
</global-modules>
```

- h. update this element from

```
<inet-address value="{jboss.bind.address.management:127.0.0.1}"/>
```

to

```
<inet-address value="{jboss.bind.address.management:<jbossHostIP}"/>
```

- i. update this element from

```
<socket-binding name="http" port="8080"/>
```

to

```
<socket-binding name="http" port="<TrueviewPort>"/>
```

6. Modify \$JBOSS_HOME/bin/standalone.conf

- a. insert this line

```
JAVA_HOME="<java_homeDir>"
```

- b. insert this line

```
JAVA_OPTS="-D[Server:server-one] -XX:PermSize=256m -XX:MaxPermSize=256m -Xms1303m
-Xmx1303m -server -Djboss.bind.address=<JBoss svr IP> -
Dsun.rmi.dgc.client.gcInterval=3600000 -Dorg.jboss.resolver.warning=true -
Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true -
Djboss.modules.system.pkgs=org.jboss.byteman -Djava.net.preferIPv4Stack=true -
Dserverid=<JBoss svr IP> -Djboss.bind.address.management=<JBoss svr IP> -
Dsun.lang.ClassLoader.allowArraySyntax=true -
DlicenseFile=<Trueview_appDir>/license/<license>.lic"
```

2.4.2.5 Trueview Startup/Shutdown

1. Startup
 - a. As user jboss:

```
>> $JBOSS_HOME/bin/standalone.sh
```



NOTE: Monitor startup progress in `$JBOSS_HOME/bin/logs/tnp.log`. Check for “Trueview application server: Started”.

2. Shutdown
 - a. As user jboss:

```
>> ps -ef |grep standalone
>> kill <standalone pids>
```



NOTE: Monitor shutdown progress in `$JBOSS_HOME/bin/logs/tnp.log`. Check for “Shutdown all connectors”.

2.5 Configuring Trueview for Event Collection

2.5.1 Collector Configuration

If the environment is making use of Event Management, then it is typical that the events are coming from an external source. In order to get these events into TNP, or more specifically EVM, it is necessary to setup a collector to retrieve the events and transfer them into the Event Manager (EVM).

The configuration of a collector relies on EVM having been setup and running to ensure successful setup.

First, setup a collector directory within the environment:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>
$ mkdir collector
$ cp -p release/samples/collector-config.properties collector/
$ cd collector
Open the file collector-config.properties for editing and make the following
adjustments:
COLLECTORID = <collectorid>
CORE-URL = http://
://<adminservername>:<evmwebobjectport>/evmhttp/eventSubmitServlet
WEBSERVICE-PORT=<webserviceport>
WEBSERVICE-ENABLED = true
WEBSERVICE-DATAMAP = <webservicedatamap>
WEBSERVICE-IGNORE-TID-CASE = true
```

The `<collectorid>` is just a name to assign to the collector while the `<webserverport>` is any valid port that can be used on the local host. The `<webservicedatamap>` is something that is defined by the environment’s external event provider.

Finally, the TNP database needs to be updated to include the id of the collector (`<collectorid>`) that has been created. Using any valid SQL tool, connect to the database `<dbtablespaceTNP>` and execute the following:

```
insert into collector values ('<collectorid>', 'Collector for <EventSourceName>',
'<EventSourceProtocol>');
```

2.5.2 Collector Wrapper Scripts

As the typical that the user environment may not have all the correct settings, it is recommend to create wrapper scripts for starting and stopping the collector.

Here are examples of the two scripts that are recommended to be created within the collector's home directory

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>/collector
```

Open the file RunCollector.sh for editing and add the following lines (adjusting as appropriate)

```
#!/bin/sh
# The JAVA_HOME and TNP_HOME variables need to be set
# either in the environment or by uncommenting these entries
TNP_HOME=<Trueview_appDir>
export TNP_HOME
JAVA_HOME=<java_homeDir>
export JAVA_HOME
${TNP_HOME}/bin/runCollector.sh $@
```

Open the file StopCollector.sh for editing and add the following lines (adjusting as appropriate)

```
#!/bin/sh
# The JAVA_HOME and TNP_HOME variables need to be set
# either in the environment or by uncommenting these entries
TNP_HOME=<Trueview_appDir>
export TNP_HOME
JAVA_HOME=<java_homeDir>
export JAVA_HOME
${TNP_HOME}/bin/stopCollector.sh $@
```

2.6 Configuring Connectors

2.6.1 Alcatel SAM Connector

The Alcatel SAM connector is configured by creating the following XML in TNP GUI > Configuration > Connectors > SAM-Connector

```
<?xml version="1.0" encoding="UTF-8"?>
<TNPConfiguration>
  <tnp-connectors-configuration>
    <connector-libraries>
      <connector-library lib="connector.alcatel5620SAMO.jar">
        <connector-instances>
          <connector-instance name="Alcatel 5620 SAM-O Connector"
autopopulate="true">
            <init-parameters>
              <parameter name="url" value="<SAM_URL>" />
              <parameter name="username" value="<SAM_userid>" />
              <parameter name="password" value="<SAM_encrypted_pw>" />
              <parameter name="connectionTimeout" value="120000" />
              <parameter name="operationTimeout" value="100000" />
              <parameter name="uploadXMLResponseDirectory"
value="<SAM_XML_DIR>" />
            </init-parameters>
          </connector-instance>
        </connector-instances>
      </connector-library>
    </connector-libraries>
  </tnp-connectors-configuration>
```

```
</TNPConfiguration>
```

The encrypted password is generated with MD5 ENCRYPTION TOOL <http://www.stringfunction.com/md5-hash.html>

2.6.2 ANDA Connector

The ANDA connector is configured by creating the following XML in TNPGUI > Configuration > Connectors > ANDA-Connector

```
<?xml version="1.0" encoding="UTF-8"?>
<TNPConfiguration>
  <tnp-connectors-configuration>
    <connector-libraries>
      <connector-library lib="connector.ANDA.jar">
        <connector-instances>
          <connector-instance name="ANDA Connector" autopopulate="true">
            <init-parameters>
              <parameter name="username" value="<ANDA_userid>" />
              <parameter name="password" value="<ANDA_pw>" />
              <parameter name="connectionTimeout" value="120000" />
              <parameter name="operationTimeout" value="100000" />
            </init-parameters>
          </connector-instance>
        </connector-instances>
      </connector-library>
    </connector-libraries>
  </tnp-connectors-configuration>
</TNPConfiguration>
```

Chapter 3

Trueview Upgrade

3.1 Install Trueview Files

1. Log into the Trueview user account on the application server.
2. Copy the application release contents from the physically shipped CD or extract the content from the digitally shipped zipped file into the previously created Trueview app directory. There are a number of files and directories included in the distribution. Please ensure that all files are transferred and/or extracted.

If physical CD, first place the CD into a reachable device for the host to read, then

```
$ scp -pr <CD>:* ./
```

If digital file, first transfer the file(s) into the current directory, then

```
$ gunzip <digitalfile>.zip
$ tar xf *.tar
```

Upon completion there will be some or all of these directories:

- tnp-<version#>-<build#>
- CustomEventHandlers
- Custom

3. Backup the database.

For Oracle, perform the backup via the Oracle administration UI, or login into an appropriate host as **oracle** and make use of the exp or expdp tools. Running them on the command line without parameters will put them into interactive mode. Simply answer the questions as presented.

For PostgreSQL, log in as **postgres** user on a server where PostgreSQL is running and dump table to a file, as follows:

```
$ pg_dump tnp_serverport >
~postgres/<dbtablespace>_<ReleaseNum>_<YearMonthDay>.sql
```

If logged in as **oracle** or **postgres** user, then log out.

4. Login as the web server user (weblogic or jboss) as Trueview needs to be re-configured by the web server user.
5. Backup the current Trueview installation.

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>
$ tar cf ../<tnpwebobjectname>_<ReleaseNum>_<YearMonthDay>.tar ./
$ gzip ../<tnpwebobjectname>_<ReleaseNum>_<YearMonthDay>.tar
```

For convenience, additionally back up the adapters and reports to a different directory:

```
$ cp -rp adapters adapters_<YearMonthDay>
$ cp -rp reports reports_<YearMonthDay>
```

6. Stop the application.

If using a WebLogic webserver, either use the Administration UI to stop the managed server <tnpwebobjectname> or use the script `stopManagedWebLogic.sh`.

If using JBoss webserver, it is necessary to stop the web server instance and the JBoss environment. This can be accomplished via the following:

```
$ ps -ef |grep standalone
$ kill <standalone pids>
```



NOTE: Server shutdown may take a while, but one can check log files for completion by watching the log files and waiting for the shutdown message.

```
$ tail -f <webserver_homeDir>/tnp-config/<tnpwebobjectname>/logs/tnp.log
```

The shutdown message should look as follows:

```
... com.tieroneoss.tnp.connector.TR1ConnectorInstance - Shutdown all connectors.
```

7. Create a link to the new Trueview installation directories and all their contents by removing and updating the one link:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>
$ rm release
$ ln -s <Trueview_tarDir>/tnp-<version#>-<build#> release
```

8. Update the contents of the adapters and reports directories.

```
$ cp -rp release/adapters ./
$ cp -rp release/reports ./
```

9. Update the contents of the eventhandling directory. If appropriate, update the custom handlers as well:

```
$ cp -rp release/eventhandling ./
$ cp -p <Trueview_tarDir>/CustomEventHandlers/* eventhandling/
```

10. If a new license is required to run this updated version of Trueview, then first copy the received license file into the previously created Trueview tar directory (<Trueview_tarDir>) before copying it to the appropriate location.

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>/license
$ cp <Trueview_tarDir>/<tnplicense> ./
```

11. If a custom CSS file exists for a Custom User Interface, entries must be added to the end of this file to support the new Date/Timestamp widget in 9.1:

```
/*GWT DateBox and DatePicker*/
.gwt-DatePicker {
  border: 1px solid #ccc;
  border-top: 1px solid #999;
  cursor: default;
}
```



```
.gwt-DatePicker td,  
.datepickerMonthSelector td:focus {  
    outline: none;  
}  
  
.datepickerMonthSelector td:focus {  
    outline: none;  
}  
  
.datepickerDays {  
    width: 100%;  
    background: white;  
}  
  
.datepickerDay,  
.datepickerWeekdayLabel,  
.datepickerWeekendLabel {  
    font-size: 85%;  
    text-align: center;  
    padding: 4px;  
    outline: none;  
    font-weight: bold;  
    color: #333;  
    border-right: 1px solid #EDED;EDED;  
    border-bottom: 1px solid #EDED;EDED;  
}  
  
.datepickerWeekdayLabel,  
.datepickerWeekendLabel {  
    background: #fff;  
    padding: 0px 4px 2px;  
    cursor: default;  
    color: #666;  
    font-size: 70%;  
    font-weight: normal;  
}  
  
.datepickerDay {  
    padding: 4px 7px;  
    cursor: hand;  
    cursor: pointer;  
}  
  
.datepickerDayIsWeekend {  
    background: #f7f7f7;  
}  
  
.datepickerDayIsFiller {  
    color: #999;  
    font-weight: normal;  
}  
  
.datepickerDayIsValue {  
    background: #d7dfe8;  
}  
  
.datepickerDayIsDisabled {  
    color: #AAAAAA;  
    font-style: italic;  
}  
  
.datepickerDayIsHighlighted {
```

```

background: #F0E68C;
}

.datePickerDayIsValueAndHighlighted {
background: #d7dfe8;
}

.datePickerDayIsToday {
padding: 3px;
color: #fff;
background: url(../hborder.png) repeat-x 0px -2607px;
}

.datePickerMonthSelector {
width: 100%;
padding: 1px 0 5px 0;
background: #fff;
}

.datePickerPreviousButton,
.datePickerNextButton {
font-size: 120%;
line-height: 1em;
color: #3a6aad;
cursor: hand;
cursor: pointer;
font-weight: bold;
padding: 0px 4px;
outline: none;
}

td.datePickerMonth {
text-align: center;
vertical-align: middle;
white-space: nowrap;
font-size: 100%;
font-weight: bold;
color: #333;
}

.gwt-DateBox {
/*padding: 5px 4px;
font-size: 100%;*/
border: 1px solid #ccc;
border-top: 1px solid #999;
}

.gwt-DateBox input {
width: 8em;
}

.dateBoxFormatError {
background: #ffcccc;
}

.dateBoxPopup {
}
/*END GWT DateBox and DatePicker*/

```

12. Execute the database script to update the database content and clear the stored checksums:

```
$ cd <webserver_homeDir>/tnp-config/<tnpwebobjectname>/database_scripts
$ ./databaseUpdate.sh --clear
```

You will be prompted to enter database parameters. The following is the console output of the script:

```
Do you want to continue? (y/n)
y
Please enter the database type for this installation (oracle, postgres):
oracle or postgres
Please enter database user name:
<dbusername>
Please enter database user password:
<dbpassword>
Please enter the database server host name or ip:
<localhost> or <IPaddr> or <dbhostname>
Please enter the database server port(5432):
<dbserverport>
Please enter the tablespace name for data:
<dbtablespace>
Please enter the tablespace name for indexes:
<dbtablespace>
Please enter the LDAP authentication scheme for this installation (openldap,
activedirectory): openldap
Successful output will look something like this:
Updating Change Set Checksums ...
Migration successful
Updating Application Schema ...
Migration successful
Updating Application Privileges ...
Migration successful
```



NOTE: Since all required Trueview database tables are being created by these scripts, the total running time might be close to 2-4 minutes. Confirm that three “Migration successful” messages are returned from the update script as it runs. If any one step is unsuccessful, the other will not run – contact HEWLETT PACKARD ENTERPRISE Support for assistance with any error messages on installation/configuration.

3.2 Update the Deployed Applications

3.2.1 WebLogic

With the change in the release directory structure it is necessary to update the deployments within WebLogic. This is accomplished through the WebLogic administration UI. Hence, it is necessary for the WebLogic administration server to be running and accessible as well as the associated Node Manager.

1. Log into the WebLogic Admin Console using <webadminusername> and <webadminpassword>.
2. Under the ‘Domain Structure’ frame select Deployments. Click the check box beside the first web-service application and click the Update button. The following diagram reflects what the frame will look like:

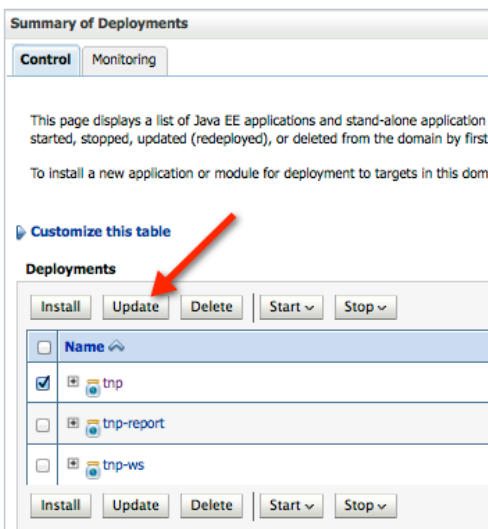


Figure 4 WebLogic deployments

3. On the next screen, confirm that the path is correct as `<webserver_homeDir>/tnp-config/<tnpwebobjectname>` and adjust if necessary via the Change Path button. The following diagram reflects what the frame will look like:

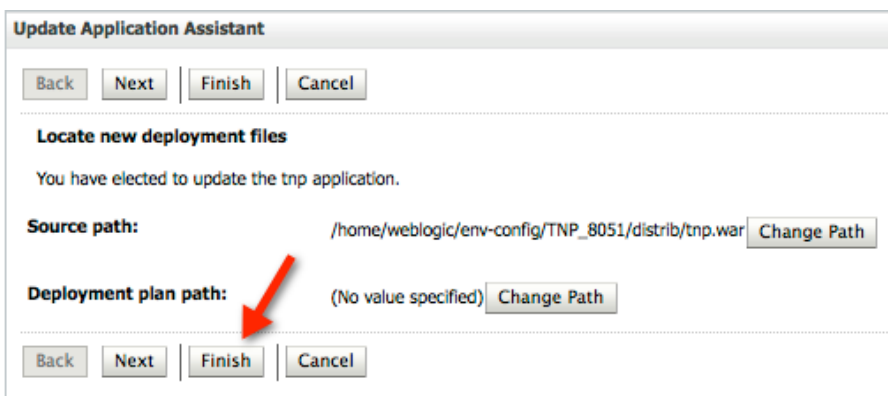


Figure 5 WebLogic update application assistant

4. Click the Finish button to complete the update process.
5. Repeat the previous steps for each Trueview specific deployment. The list should include the following objects:
 - a. tnp
 - b. tnp-ws
 - c. tnp-report
 - d. tnp-custom (if applicable)
 - e. evm (if applicable)
6. Restart the Trueview managed server `<tnpwebobjectname>` from either the WebLogic Administration UI or use the script `stopManagedWebLogic.sh`.

3.2.2 JBoss

With the new release it is not necessary to replace any existing files within the JBoss environment because the war files are linked. Simply, restart the JBoss environment to pick up the new war files:

```
$ $JBOSS_HOME/bin/standalone.sh
```

Monitor startup progress in `$JBOSS_HOME/bin/logs/tnp.log`. Check for “Trueview application server: Started”.

Chapter 4

Verification of signed binary

This Software Product from Hewlett Packard Enterprise is digitally signed and accompanied by Gnu Privacy Guard (GnuPG) signatures. Hewlett Packard Enterprise strongly recommends using signature verification on its products, but there is no obligation. Customers will have the choice of running this verification or not as per their IT Policies.

4.1 Installing and Configuring Gnu Privacy Guard (GnuGP)

If you do not already have GnuGP installed, you will first need to download and install it. For information about obtaining and installing GnuGP, see <http://www.gnupg.org>.

Before verifying the signatures delivered on the HPE Service Activator DVD, you need to configure GnuGP for accepting signatures from Hewlett Packard Enterprise. To do this, follow these steps:

1. Log on your system.
2. Get the all public keys from following location:
<https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPLinuxCodeSigning>
3. Save the keys as `HP-RPM-GPG-1024-KEY.pub`, `HP-RPM-GPG-2048-KEY.pub`, `HP-RPM-GPG-2048-KEY-1.pub`, `HP-RPM-GPG-2048-KEY-2.pub`, `HP-RPM-GPG-2048-KEY-3.pub`, `HP-RPM-GPG-2048-KEY-4.pub`, and `HP-RPM-GPG-2048-KEY-5.pub`, respectively.
4. Import the key into GnuPG by running these commands:

```
gpg --import HP-RPM-GPG-1024-KEY.pub
gpg --import HP-RPM-GPG-2048-KEY.pub
gpg --import HP-RPM-GPG-2048-KEY-1.pub
gpg --import HP-RPM-GPG-2048-KEY-2.pub
gpg --import HP-RPM-GPG-2048-KEY-3.pub
gpg --import HP-RPM-GPG-2048-KEY-4.pub
gpg --import HP-RPM-GPG-2048-KEY-5.pub
```

4.2 Verifying the Authenticity and Integrity of the Software

The procedures listed below allow you to assess the integrity of the software before installing it, by verifying the signatures of the software packages.

From a command prompt, go to the `/Binaries` directory on the DVD and run the following command:

```
gpg --verify trueview-8.0-3441.tar.gz.sig trueview-8.0-3441.tar.gz
```

Look for the following output from the `gpg` command:

```
gpg: Good signature from "Hewlett-Packard Company RSA (HP Codesigning Service) - 2"
```