ATAF Installation

ATAF Version: v0.5 Tech18

Created Data: 28 Apr 18

Author: Simon Hunt

# Support Versions

|  |  |  |
| --- | --- | --- |
| App | Version | Notes |
| Apex | 18.2 |  |
| ORDS | 3.0 | Not required for standalone use only |
| Database | 12c | XE and 11g should also work. |

# Notes

This is a full installation only and not an upgrade, please do not try and install it over the top of the ATAF Alpha release. Moving forward, I will endeavour to produce incremental patches.

Identify a target schema and workspace. ATAF now uses agents to return UI maps from target applications and parsing schemas; it can therefore exist in its own schema and workspace.

During the installation process a log file is generated in the staging directory. Some views cross reference each other and therefore show as invalid during the installation. These should recompile ok once the build is complete.

# Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Description** | **Example** |
| #ATAF\_Schema# | Schema where ATAF database will be installed | ATAF |
| #ATAF\_Workspace# | Workspace where the application will be installed | ATAF |
| #ATAF\_App\_No# | Available Apex application number | 242 |
| #ATAF\_Pwd# | Password for the ATAF\_Schema | AtaffyTafTaf2! |
| #ATAF\_Staging# | Staging folder for the source code | c:/ataf |
| #Service\_Name# | Service Name for the DB ATAF is be installed to | XE |
| #Target\_Schema# | Schema for application(s) to be tested | SAMPLE\_APP |
| #Target\_Schema\_pwd# | Schema Password for application(s) to be tested | Sample1! |
| #Wallet\_Path# | The Path to the Oracle Wallet | file:/etc/ORACLE/WALLETS/wallet |
| #Wallet\_Password# | The Password for the Oracle Wallet | 12Passw0rdxyz |

1. Download Source. From Git Hub download the latest version of ATAF to #ATAF\_Staging#
2. Connect to ATAF Schema.

sqlplus #ATAF\_Schema#/#ATAF\_Pwd#@#Service\_Name#

1. Run Install script.

@#ATAF\_Staging#/install.sql #ATAF\_Workspace# #ATAF\_App\_No#

# Post installation tasks

Background. ATAF produces a UI Map of target applications using a series of UNION ALL queries build from a number of the Apex Views. In order to make ATAF performant, the UI Map is a single Materialized View that can be set to refresh out of hours. Unfortunately, when running Apex Views as a db user, it will only show components for which it is the parsing schema. This limitation means that the UI Map views need to be deployed into each parsing schema for target applications to be tested.

## Set MV Refresh times

By default, the MVs will refresh each day at the same time they were created. Ideally, these should be changed to refresh out of hours.

## **Enable Web Services**

ATAF can run scripts locally using the Selenium IDE or on a Cloud Selenium Grid via Web Services. In order to allow these features to run, you will need to add the Wallet Path and Password to the System Parameters table. Depending on your configuration, it may also be necessary to add the testingbot url (https://api.testingbot.com) to the access control list. To update the Wallet details run the following as the ATAF owning user:

UPDATE ataf\_system\_parameters

SET value = '#Wallet\_Path#',

WHERE name = 'Wallet Path';

UPDATE ataf\_system\_parameters

SET value = '#Wallet\_Password#',

WHERE name = 'Wallet Password'

## Install ATAF Agents

In order for ATAF to see the UI components for target applications to test, an agent must be installed into the parsing schema of the target application. If ATAF is installed into the parsing schema for target application then no further action is required.

Alternatively, manually refreshing the UI map from within ATAF will include all UI items for all applications in the logged-in workspace. Automatic MV refresh will only include items for which the MV owner is the parsing schema. Therefore, if you are happy to manually refresh the MVs each time you use ATAF then the following is not required.

1. From the ATAF Schema, grant access to the UI Map view

grant select on ATAF\_APEX\_PAGE\_ITEMS\_V to #Target\_Schema#

1. Connect to Target Schema

connect #Target\_Schema#/#Target\_Schema\_Pwd#

1. Install Agent

@#ATAF\_Staging#/MATERIALIZED\_VIEWS/ATAF\_AGENT\_PAGE\_ITEMS\_MV.sql

@#ATAF\_Staging#/MATERIALIZED\_VIEWS/ATAF\_AGENT\_PAGES\_MV.sql

@#ATAF\_Staging#/MATERIALIZED\_VIEWS/ATAF\_AGENT\_SHORTCUTS\_MV.sql

@#ATAF\_Staging#/MATERIALIZED\_VIEWS/ATAF\_AGENT\_THEMES\_MV.sql

1. Update the ATAF\_AGENT\_PAGE\_ITEMS\_MV to reference the ATAF Page Items View and refresh it. Instructions for how to update the MV are commented in the SQL.
2. Give ATAF access to the Agents

GRANT select ON ATAF\_AGENT\_PAGE\_ITEMS\_MV to #ATAF\_Schema#

/

GRANT select ON ATAF\_AGENT\_PAGES\_MV to #ATAF\_Schema#

/

GRANT select ON ATAF\_AGENT\_SHORTCUTS\_MV to #ATAF\_Schema#

/

GRANT select ON ATAF\_AGENT\_THEMES\_MV to #ATAF\_Schema#

/

1. Update ATAF views to reference the Agent views. Instructions and examples for these updates can be found in the sql code. The following Views are to be updated:

ATAF\_APEX\_PAGE\_ITEMS

ATAF\_APEX\_PAGES\_V

ATAF\_APEX\_SHORTCUTS\_V

ATAF\_APEX\_THEMES\_V

# Logging into ATAF

ATAF uses workspace user authentication and therefore all your existing workspace users will have access to ATAF. The installation process creates a new group in the workspace call ataf\_administration the can be used to give administration access to user. Two example users are created during the installation process, which are used in the example ATAF scripts that are created during the installation. If you change the passwords or delete them, then you will need to amend the User data in the example ATAF scripts.

Two users are created during the installation process:

ataf\_admin / Password1!

ataf\_basic / Password1!