OIC – Replace DB Adapter With ATP Guide

Table of Contents

[Notional Architecture 2](#_Toc53385588)

[Disclaimer 3](#_Toc53385589)

[Script assumptions and notes 4](#_Toc53385590)

[How to use the automation script 4](#_Toc53385591)

[Adapter Replacement Instructions 6](#_Toc53385592)

# Notional Architecture

This guide provides a complete instruction on how to replace DB adapters with ATP adapters in existing OIC integrations, including example shell scripts that automates the steps of the replacement in the configuration files.

**Diagram

Description automatically generated**

# Disclaimer

**<INSERT THE DISCLAIMER HERE>**

# Script assumptions and notes

1. Install Python 3 to run this script.
2. The DB .iar files, the script, and unzipped ATP adapter .iar file are in the same directory
3. This script is developed on Windows
4. The .iar files should be extracted in a folder which the script will read. This means - extract packages files and export all integrations file that contain zipped iar files.
5. The user is free to edit and use sections of the below script at their own discretion
6. The user will perform testing after applying the script

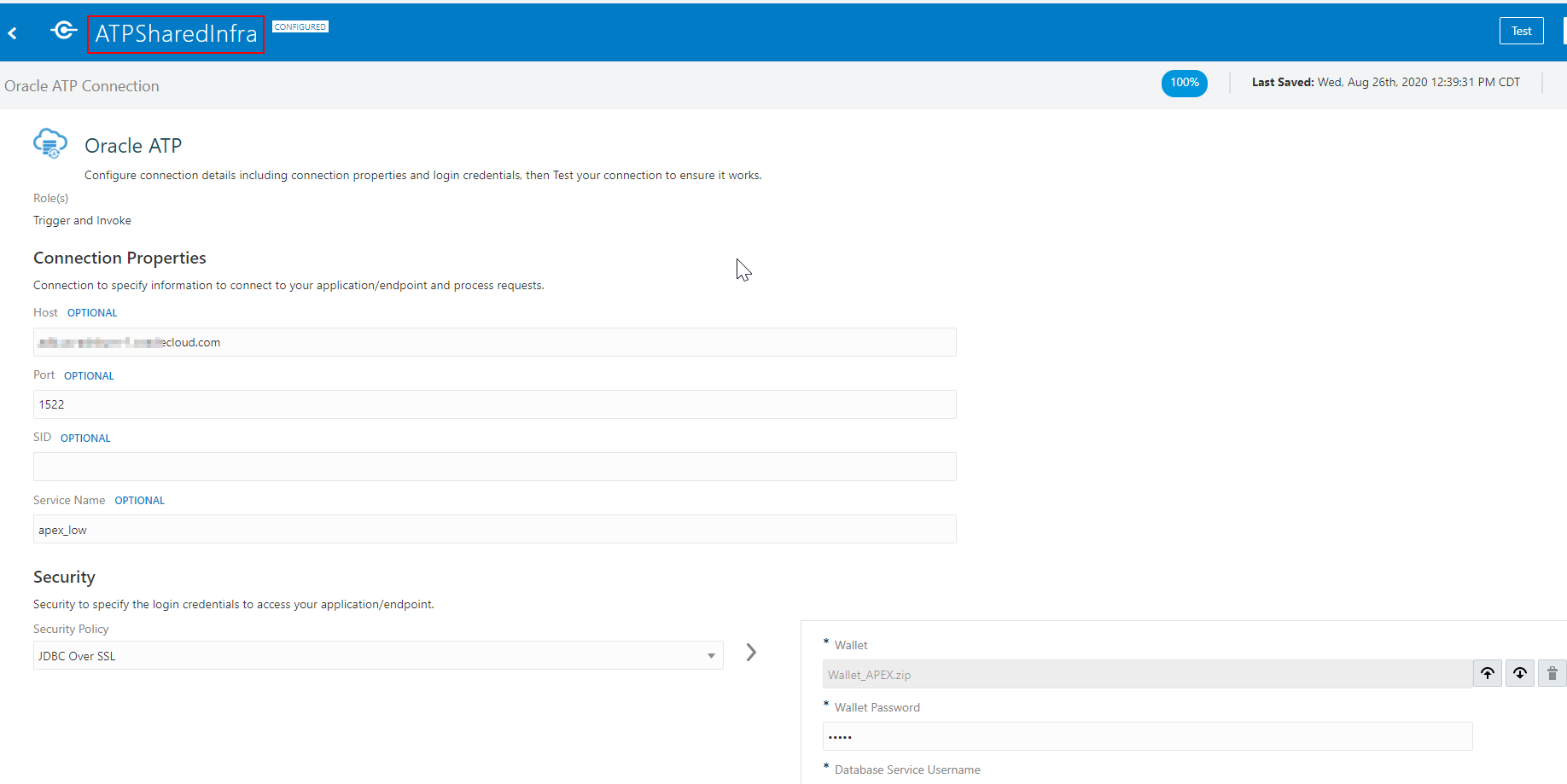
## How to use the automation script

1. Export the integrations from OIC as a .iar files in a directory
2. Create a python file and follow the instructions to create the automation script.
3. Extract all the necessary files, including ATP adapter config.
4. Edit the “Variables” section and set the variables with correct path and identifiers.
5. Run the script in python using an IDE or terminal. ex in terminal type python ReplaceDBAdapter.py
6. Import the results of the script in the new directory to OIC
7. Test your OIC connection

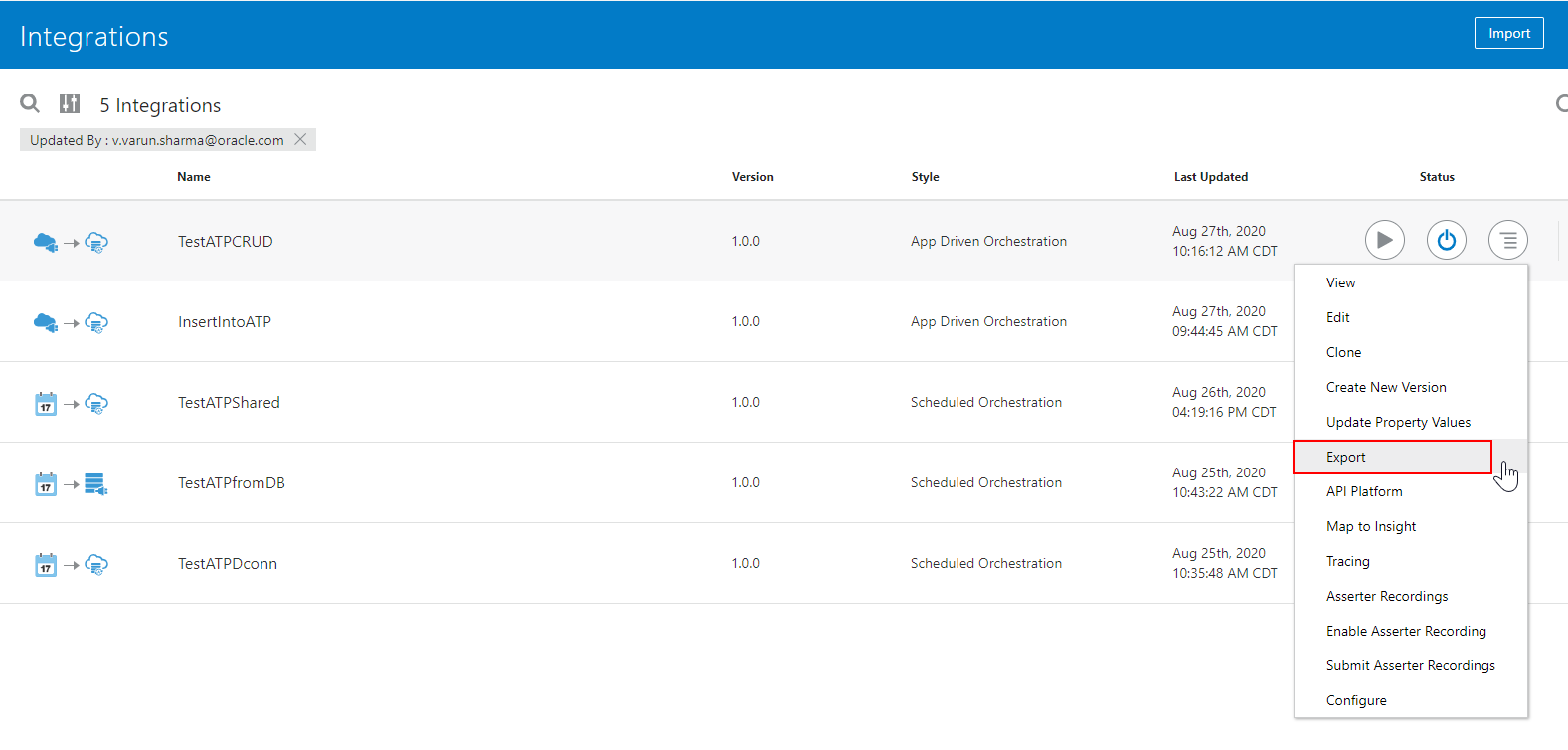
**Python Script to Replace DB Adapter with ATP guide**

**Create integration with new ATP Adapter**

1. Create a new ATP adapter and configure the connection. (ATPSharedInfra in this example)



1. Create a test integration - TestIntegration using the ATP connector, so that we can export it to get the connection details.
2. Export the integrations to get DB and ATP connection files.

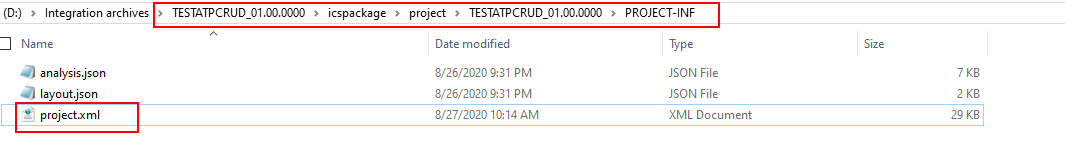


1. Unzip the exported .iar file.

**Write Script to Replace DB Adapter with ATP**

Create a script to read a directory containing the archive files, and replace the adapter configurations in them. The script will create the new iars in a separate folder.

1. Define the variables
2. **import** os
3. **import** shutil
4. **import** re
5. **import** zipfile
6. **from** xml.dom **import** minidom
8. #####################################################################################
9. #           Define these variables before running the script
10. #####################################################################################
11. #
12. #     DATABASEADAPTERNAME    =    The identifier of the current database adapter
13. #     ATPADAPTERNAME         =    Identifier of the new ATP Adapter
14. #     ATPADAPTERLOCATION     =    the path to the new ATP adapter config file (~.xml)
15. #     ATPADAPTERJCAFILE      =    Path to the JCA file for the new ATP adapter. Can be found in the new exported integration made with ATP adapter
16. #     ZIPDIRECTORY           =    Path to the directory where all the archive files(.iar) are placed. With the folder name as the last element in path e.g. e.g. D:/mydir/temp
17. #     EXTRACTTODIRECTORY     =    Path to the new directory where the script will place the updated archives. Should end with the folder name e.g. D:/mydir/temp
18. #
19. ####################################################################################
21. DATABASEADAPTERNAME = 'TESTATPFROMDB'
22. ATPADAPTERNAME = 'ATPSHAREDINFRA'
24. ATPADAPTERLOCATION = 'D:/Integration archives/TESTATPSHARED\_01.00.0000/icspackage/appinstances/ATPSHAREDINFRA.xml'
25. ATPADAPTERJCAFILE = 'D:/Integration archives/PythonScripts/TestATP\_S\_REQUEST.jca'
27. ZIPDIRECTORY = 'D:/Integration archives/PythonScripts/IARFILES'
28. EXTRACTTODIRECTORY = 'D:/Integration archives/PythonScripts/NewFolder'
30. ## Namespace variables.
31. namespaceOld = "http://xmlns.oracle.com/cloud/adapter/database"
32. namespaceNew = "http://xmlns.oracle.com/cloud/adapter/atpdatabase"
33. ics\_project\_attributes = ':atpdatabase'
34. dir\_path = os.path.dirname(os.path.realpath(\_\_file\_\_))
35. Write the function to extract the iar files(unzip and zip again for later).
36. #-----------------------------------------------------------------------
37. # Function to Extract zip into a target directory
38. #-----------------------------------------------------------------------
39. **def** extractIARFiles(path\_to\_zip\_file):
40. with zipfile.ZipFile(path\_to\_zip\_file, 'r') as zip\_ref:
41. zip\_ref.extractall('targetdir')
43. #-----------------------------------------------------------------------
44. # Function zip back the updated files to .iar
45. #-----------------------------------------------------------------------
46. **def** zipFileAndExport(zipName):
47. shutil.make\_archive(EXTRACTTODIRECTORY + '/' + zipName, 'zip', dir\_path + '/targetdir')
48. shutil.rmtree(dir\_path + '/targetdir')
49. os.rename(EXTRACTTODIRECTORY + '/' + zipName + '.zip', EXTRACTTODIRECTORY + '/' + zipName)
50. Write the function to replace the DB adapter with ATP adapter.
51. #-----------------------------------------------------------------------
52. # Function to Delete DB adapter configuration file and replace with ATP
53. #-----------------------------------------------------------------------
54. **def** replaceDBAdapterWithATP():
55. FileToBeRemoved = DATABASEADAPTERNAME + '.xml'
56. os.remove(dir\_path + '/targetdir/icspackage/appinstances/' + FileToBeRemoved)
57. tail = os.path.split(ATPADAPTERLOCATION)
58. shutil.copy(ATPADAPTERLOCATION, dir\_path + '/targetdir/icspackage/appinstances/' + tail[1])
59. Write the function that will make changes in the project.xml file.
60. #-----------------------------------------------------------------------
61. # Function to Update project.xml
62. #-----------------------------------------------------------------------
63. **def** updateProjectXML(zipName):
64. filepath = dir\_path + '/targetdir/icspackage/project/' + zipName.rpartition('.')[0] + '/PROJECT-INF/project.xml'
65. #print(filepath)
66. substring = '<code>' + DATABASEADAPTERNAME + '</code>'
67. newstring = '<code>' + ATPADAPTERNAME + '</code>'
69. changeString(filepath, substring, newstring)
70. changeString(filepath, namespaceOld, namespaceNew)



**Replace Mapper**

1. Make sure you have set the correct path to the connection properties for the ATP adapter. The following function will change all the jca files with ATP config.
2. #-----------------------------------------------------------------------
3. # Function to update JCA files to edit config for ATP, and change namespaces.
4. # Calls changeJCAProperties
5. # - Args needed file path for JCA file, and file path for ATP JCA File
6. #-----------------------------------------------------------------------
7. **def** updateJCAFilesAndNamespaces(zipName):
8. prjctdir = dir\_path + '/targetdir/icspackage/project/' + zipName.rpartition('.')[0]
9. changeString(prjctdir + '/ics\_project\_attributes.properties', ':database', ics\_project\_attributes)
10. **for** subdir, dirs, files **in** os.walk(prjctdir + '/resources'):
11. **for** file **in** files:
12. #print(os.path.join(subdir, file))
13. **if** file.rpartition('.')[2] == 'jca':
14. changeJCAFileProperties(os.path.join(subdir, file), ATPADAPTERJCAFILE)
16. changeString(os.path.join(subdir, file), namespaceOld, namespaceNew)
18. **def** changeJCAFileProperties(jcaFilePath, ATPJCAFILE):
19. ## ------ Check if jca file is for database adapter ----------------
20. mydoc = minidom.parse(jcaFilePath)
21. items = mydoc.getElementsByTagName('adapter-config')
23. **if** items[0].attributes['adapter'].value == "database":
24. ## ------ Change adapter type to atp --------------------
25. items[0].attributes['adapter'].value = 'atpdatabase'
27. ## ------ Clone connection factory from ATP JCA file ------------------------------
28. connectionStringATP = minidom.parse(ATPJCAFILE)
29. atpAdapterConn = connectionStringATP.getElementsByTagName('connection-factory')[0]
30. cloneNode = atpAdapterConn.cloneNode(deep=True)
32. # Change current JCA file properties
33. connFactNode = mydoc.getElementsByTagName('connection-factory')
34. endPointNode = mydoc.getElementsByTagName('endpoint-interaction')
36. # remove old connection-factory
37. **for** i **in** connFactNode:
38. x = i.parentNode
39. x.removeChild( i )
40. # remove endpoint
41. **for** i **in** endPointNode:
42. x = i.parentNode
43. x.removeChild(i)
45. # remove on-prem and agent attribute from endpoint-interaction
46. **if** endPointNode[0].hasAttribute('agent'):
47. endPointNode[0].removeAttribute('agent')
48. **if** endPointNode[0].hasAttribute('onpremise'):
49. endPointNode[0].removeAttribute('onpremise')
51. # add both connfactory and endpoint interaction back to xml
52. mydoc.firstChild.appendChild(cloneNode)
53. items[0].appendChild(endPointNode[0])
55. # write the file
56. f = open(jcaFilePath, "w")
57. f.write(mydoc.getElementsByTagName('adapter-config')[0].toxml())
58. f.close()

This will change the jca files for atp database.



1. Write the function to change all the namespaces.
2. #-----------------------------------------------------------------------
3. #    Function to replace a particular string in file
4. #-----------------------------------------------------------------------
5. **def** changeString(filename, substringOld, substringNew):
6. with open(filename, 'r+') as f:
7. text = f.read()
8. **if** substringOld **in** text :
9. text = re.sub(substringOld, substringNew, text)
10. f.seek(0)
11. #print(text)
12. f.write(text)
13. f.truncate()
14. Finally write the main function to run the script and create new iars in a new folder.
15. **def** main():
16. **for** filename **in** os.listdir(ZIPDIRECTORY):
17. **if** filename.endswith(".iar"):
18. **print**(os.path.join(filename))
19. extractIARFiles(os.path.join(ZIPDIRECTORY, filename))
20. replaceDBAdapterWithATP()
21. updateProjectXML(filename)
22. updateJCAFilesAndNamespaces(filename)
23. zipFileAndExport(filename)
24. **else**:
25. **continue**
27. **if** \_\_name\_\_ == "\_\_main\_\_":
28. main()

**NOTE**: This approach will only work if it’s a complete migration from Database to ATP. If both the adapters will be used in some integration, please do not do use this script. Also, make sure that all the variables are set.