



# **APEX Backup Tool Development**

**September 2015**

Author:  
Marija Pinkute

Supervisor:  
Katarzyna Maria Dzedziniewicz-Wojcik

CERNopenlab Summer Student Report 2015



## **Project Specification**

At CERN Oracle Application Express (APEX) is widely used to facilitate development of web applications based on data stored in Oracle. Oracle provides a tool that offers backup of applications developed in APEX, but for a long time it was missing some functionality that is crucial for CERN.

The main objectives of the project are to conduct a test of the available tool and depending on the test results, expand the tool to provide backup of missing objects and automatize the backup process.

## Abstract

At CERN Oracle Application Express (APEX) is widely used to facilitate development of web applications based on data stored in Oracle.

The task of the project was to develop a tool for backing up the in case of data loss or corruption. The first part of the project constituted of examining available backup solutions and the ways that APEX stores data. It was concluded that there is no tool that would fulfill all the desired requirements of: exporting everything at once, dealing with two versions of APEX and would include all the attached external files.

A Java tool was developed that included all the missing functionality. Moreover, a simple wrapper was written to provide a way to restore the APEX environment from backups. Tools were successfully tested by automated tests, expert examination of complex applications restored after backup and on the largest APEX installation at CERN.

# Table of Contents

Abstract.....	3
1 Introduction .....	5
2 APEX Overview.....	5
3 Available APEX Backup Solutions.....	7
3.1 APEX 4.....	7
3.2 APEX 5.....	8
3.3 Evaluation of APEX Backup Capabilities .....	9
4 Development of Automatic Backup Tool.....	9
4.1 Workspaces Export .....	10
4.2 Applications Export .....	11
4.3 Files Export .....	11
4.4 Challenges & Concerns .....	12
4.5 Running Instructions .....	13
5 Evaluation of Backup Tool.....	14
5.1 Automated Testing.....	14
5.2 User Testing.....	14
5.3 Testing on a database .....	15
6 Importing the APEX Backups .....	15
6.1 Import Wrapper .....	15
6.2 Running Instructions .....	15
7 Conclusion .....	16
8 References.....	17

# 1 Introduction

Oracle Application Express (APEX) is a tool provided by Oracle that aims to facilitate development of web applications. These applications help manage and visualise data stored in Oracle databases. APEX widely used at CERN and currently it is available on 19 databases, 13 of which are production, they include 179 workspaces and over 200 applications. Some well known examples are Training Catalogue and Critical LHC Configuration applications. The aim of this project was to build a tool for backing up APEX applications.

The motivation behind building the APEX backup tool is that in case of data loss or corruption all applications could be restored. One of the possible scenarios when something wrong can occur is when APEX minor version upgrades are carried out (where no new schema is being created). At CERN APEX version 4 is currently used and will be upgraded to APEX version 5, when the 5.1 version is released. Applications from APEX 4 used to be backed up by backing up APEX schemas in the database. However, oracle claimed that this is not a full backup and with APEX 5 this approach is not possible.

Oracle provides two java command line utilities to export APEX components for back up automation (more in Section 3). However, they do not provide full backups, for instance the external workspace files are not included. Hence, this project aims to find out the scope of the official APEX backup tool and expand it with the functionality needed.

The solution implemented has to satisfy CERN's requirements. It is important that the tool implemented would perform the back up process online, without disrupting the database being backed up. This is because there are many critical services running on the databases and there should be no impact on their work. The tool should be able to export all the components necessary per database to recreate the exact same APEX environment from backups. All in all, the main goal of the project is to implement an all-in-one solution that would provide backup of all APEX components and would enable easy automation of the backup process.

## 2 APEX Overview

The first step to building a backup tool for the APEX framework understanding how APEX is organised and what needs to be exported in order to enable a full restore.

APEX is made up of two components a metadata repository and the APEX engine. The metadata repository stores all the definitions of all the applications and the engine is responsible for rendering and processing all the application pages. APEX is stored within the Oracle database in the form of data tables and PL/SQL scripts. Figure 1 below taken from Oracle documentation provides an overview of the architecture [1].

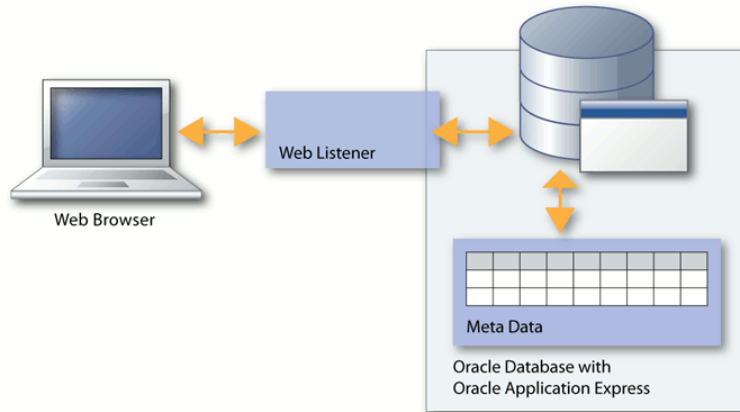


Figure 1. Official Oracle diagram showing APEX architecture overview. [1]

APEX is installed on the database that the applications will use the data from. The APEX working area is organized into workspaces that are “associated with one or more Oracle database schemas” [2]. Applications are created in workspaces. All workspaces and applications have a unique id assigned to them and are easy to track. Both workspaces and applications can have external files attached to them such as CSS, JavaScript, images, etc. Files attached to a workspace are accessible by any application in the same workspace, files attached to an application are only accessible by the application. Information about workspaces, applications and files is stored in the metadata tables described.

In terms of what needs to be included in the backup export is all the metadata on workspaces and applications (illustrated as definition files in the diagram below) and the external static files that have been attached.

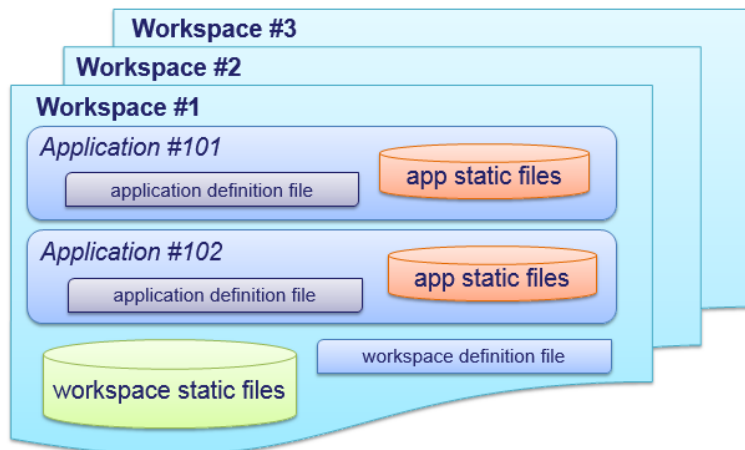


Figure 2. Diagram depicting an overview of the APEX component structure.

### 3 Available APEX Backup Solutions

The APEX backup process is to be carried out by exporting components as SQL import scripts. This is the acceptable way of backup as APEX provides web tool and command line interfaces to carry out exports in this particular way. Moreover, it ensures an easy way of import, when restoring data from the exports is needed. The following section investigates what tools exist and what components they export in detail.

As mentioned above the APEX web interface provides a manual way of exporting workspaces, applications, files, etc. Most of these exports produce an SQL script that can be later imported to an APEX installation. Using this methodology all essential components could be backed up. However, it is a manual process that cannot be easily automated.

Oracle provides two Java utility classes in the APEX installation package that can be used to export all workspace and application import scripts. The tool has to be used separately for each type of export. This package is available for both versions of APEX. The two tools are [3]:

- **APEXExport.** This java class enables one to export SQL import scripts of chosen workspace, application or all workspaces, applications. (Launch parameters include database details (DB name, user name, password) and parameters of what you want to export (for example: “-instance”, “-expWorkspace”, “-workspaceid”, “-applicationid”))
- **APEXExportSplitter.** This java class enables one to split an exported SQL script to multiple SQL scripts for different components.

Even though these two tools serve the same purpose in both versions of APEX the exports have different scope of what components they include. This is because a significant amount of changes was introduced in the release of APEX 5 that include differences in file storage. The following two subsections discuss both versions in detail.

#### 3.1 APEX 4

In APEX all the files on record can be found in APEX file views in the metadata. In version 4 all the files can be found in two views, see Table 1.

APEX View	Dictionary Explanation
APEX_WORKSPACE_FILES	Identifies uploaded files belonging to the workspace in the modplsqli or EPG documents table.
APEX_APPL_PLUGIN_FILES	Stores the files like CSS, images, javascript files, ... of a plug-in.

Table 1. This table lists and explains all the file views in APEX 4.

In APEX 4 all the external files attached to the applications and workspaces (images, CSS, JavaScript...) can be found in the database in the APEX\_WORKSPACE\_FILES view. If the APPLICATION\_ID is set to ‘0’ then it’s considered a workspace file, otherwise it’s an application one. The third party files in the web tool are split to images, CSS and static files, in the view they are only identified by FILE\_TYPE and this file type is not relevant when retrieving the file for display. All files are referenced by the WORKSPACE\_ID and the FILE\_NAME, thus

in theory there should not be any two files with the same name. These files are not included in any of the export capabilities of the tools provided. If an application was exported using the command line tool and then re-imported to a cleared workspace, it would have all of its external files missing, this is depicted in Figure 3. This is a clear drawback of the tool.

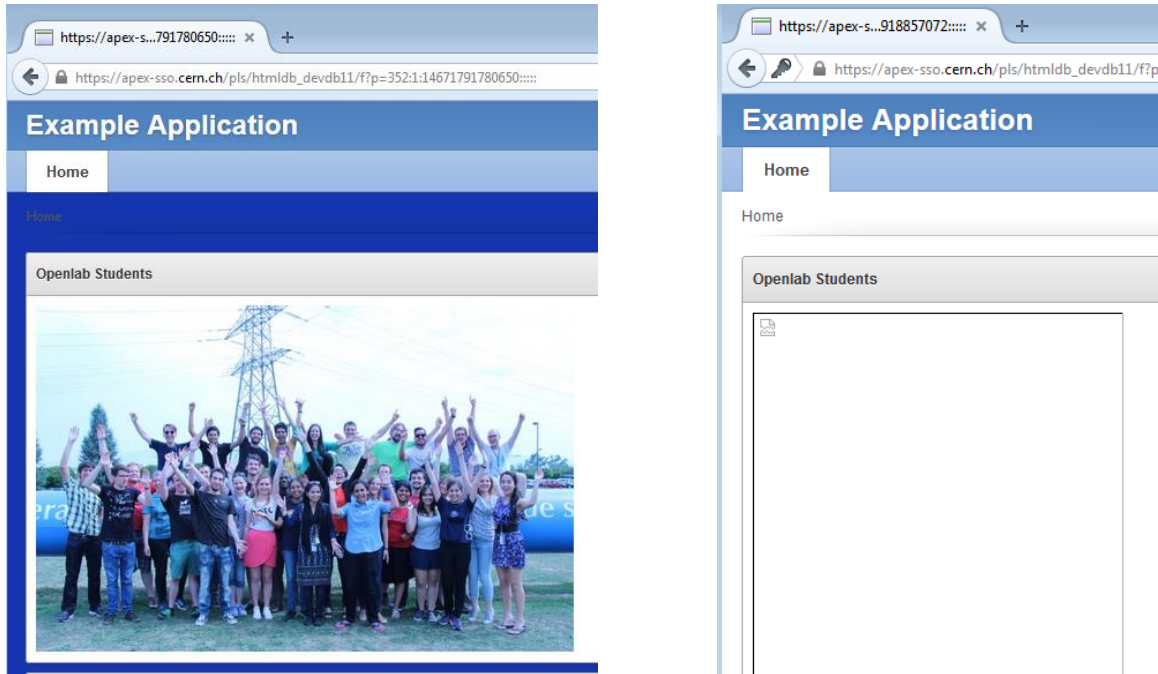


Figure 3. The application on the right is the restored from backup version of the application on the left. One can see that the image and CSS files are missing.

Other than the static files belonging to the applications or workspaces, plug in files are stored in APEX\_APPL\_PLUGIN\_FILES view. These files are linked to the plug-ins used in applications and both plug-ins and their files are exported together with the application exports.

### 3.2 APEX 5

Similarly to APEX 4 all the external files can be found in the APEX file views, however in APEX version 5 there are more of them. This is because the storage architecture was changed. All the file views can be found in Table 2.

APEX View	Dictionary Explanation
APEX_WORKSPACE_FILES	Identifies uploaded files belonging to the workspace in the modplsqli or EPG documents table.
APEX_APPL_PLUGIN_FILES	Stores the files like CSS, images, javascript files, ... of a plug-in.
APEX_WORKSPACE_STATIC_FILES	Stores the files like CSS, images, javascriptfiles, ... of a workspace.
APEX_APPLICATION_STATIC_FILES	Stores the files like CSS, images, javascriptfiles, ... of an application.



APEX_APPLICATION_THEME_FILES	Stores the files like CSS, images, javascriptfiles, ... of a theme.
APEX_APPL_CONCATENATED_FILES	Concatenated files of a user interface

Table 2. This table lists and explains all file based views in APEX 5.

Differently from APEX 4 all static files attached to workspaces and applications can be found in the view APEX\_WOKSPACE\_STATIC\_FILES and APEX\_APPLICATION\_STATIC\_FILES. The APEX\_WORKSPACE\_FILES view still exists and stores all the files as before, however in the future APEX releases will not include external static files in this view, thus one should rely on the two new views. The APEX export utility for APEX 5 includes the static application files in the application exports, but still excludes the workspace ones [4].

Files linked to APEX plug-ins used in applications are found in APEX\_APPL\_PLUGIN\_FILES, just as in APEX 4. They are also included in the application exports.

Two other file views have been added APEX\_APPLICATION\_THEME\_FILES and APEX\_APPL\_CONCATENATED\_FILES. The former view includes external static files attached to themes. These files are also included in the application exports that the themes belong to. The latter view includes concatenated files of a user interface, which are created automatically. These files do not need to be backed up.

### 3.3 Evaluation of APEX Backup Capabilities

All in all, the official backup utilities provided by APEX provide limited functionality. One can automate exporting applications and workspaces by writing a wrapper to run the tool twice (once for each case), no overall backup is possible. Moreover, both versions of the tool do not support exporting external files attached to workspaces. On the other hand, static application files are included in application exports in APEX 5, but not in APEX 4. It is clear that the same tool cannot be used for both versions of APEX used at CERN. Exporting third party files as SQL scripts that can later be used for easy import is only available in the web tool of APEX 4, APEX 5 allows only compressed zip file export.

## 4 Development of Automatic Backup Tool

As concluded in the evaluation of available APEX backup solutions (section 3.3) the solution offered by Oracle has essential functionality missing. Thus, a tool was to be built that would:

- be able to carry out the backup online, without disrupting the database
- be able to export all external files
- be able to export all applications, workspaces and files per database in one run
- be able to export external static files of all types of sizes
- be able to handle both versions of APEX (4 & 5) that are of interest to CERN

One of the APEX developers (Giovanni Chierico) at CERN had built an application backup tool for personal use. This tool was compatible with APEX 4, used the APEX official utilities for application back-up and performed workspace file export in raw and SQL import script format.

The tool also contained the possibilities to export the database, database dependencies and the functionality of back up management through SVN.

The most interesting part of this tool was the example of successful file export, because it was the essential missing part in the Oracle utility. However, thorough testing of the feature showed, that unfortunately it is unable to export very large files such as mp4 file format movies even if more memory is assigned to the JVM. Hence, the tool was used as an example, the database connection and utilities modules were re-used, the module responsible for command line parameter passing was adjusted and the unnecessary functionality was removed. The file export in SQL script format was expanded to include a header and changed so that the file would be exported incrementally avoiding any memory issues. Other functionality such as dealing with both versions of APEX, exporting everything (workspaces, applications and all relevant static files) at database level and automated testing, was implemented.

All the backed up export files are stored in a directory of the database name at the path provided. Each APEX installation exports were to organize the exports in directories: workspaces, applications, files (SQL import scripts) and raw-files (files in their original format). This simple organisation of files enables easy import later on (see Section 6).

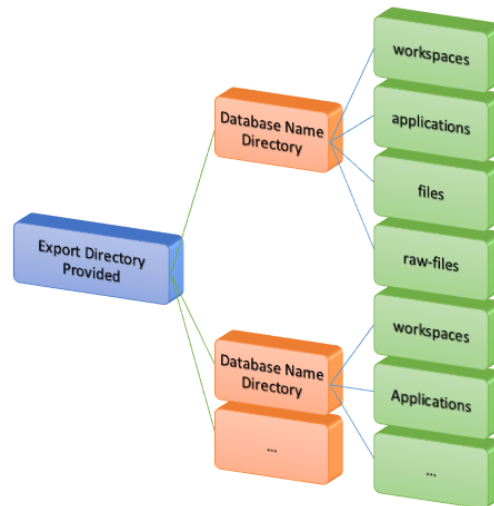


Figure 4. Diagram depicting the organisation of the backed up files.

## 4.1 Workspaces Export

The export of all workspaces relies on the official APEX tool, as it provides a reliable export. This is done by including the ApexExport.class utility in the project and running it with the parameter '-expWorkspace'. Important to note is that all the workspace export files are exported to the desired directory in the end, however during the export they appear at the location that the tool is run from. This is because the ApexExport utility does not have the functionality of providing an output location. The tool developed moves the exported files to the desired location after the utility carries out the export.

Several workspaces are not included in the export as they are linked to the APEX installation, these are two workspaces with the workspace ids '10' and '11'. Also, all the files exported have special file names for easy traceability. See table below for more information.

Workspaces excluded from export:

- ✘ workspace id: 10, workspace display name: INTERNAL
- ✘ workspace id: 11, workspace display name: COM.ORACLE.APEX.REPOSITORY

File name format:

<database name>\_w<workspace id>.sql

## 4.2 Applications Export

Similarly to workspace export the export of all application relies on the official APEX tool, as it provides a reliable export. For exporting all applications of APEX 5 the ApexExport.class is used with the '-instance' parameter, while all the APEX 4 are exported individually by running individual application exports via the '-applicationid' parameter. This is because the same export utility tool cannot be used for both versions. The method of application export is chosen depending on the version parameter passed in when launching the tool.

The same important note applies to application as for workspace exports. All the application files are exported are exported to the output directory provided, however during the export they appear at the location that the tool is run from. This is because the ApexExport utility does not have the functionality of providing an output location. The tool developed moves the exported files to the desired location after the utility carries out the export.

Three types of applications are excluded from the backup export (see table below). The APEX internal applications cannot be exported by default, other applications can be exported, however they cannot be imported, because they normally come with the installation and upon an attempt to import an error is thrown as they violate the id uniqueness constraint. Just like workspace export files, application files also have a special file name format.

### Applications excluded from export:

- ✘ application id with pattern: 4/d/d/d †-APEX internal applications
- ✘ application id with pattern: 7/d/d/d †-APEX sample applications (APEX 4)
- ✘ application id with pattern: 8/d/d/d †- theme applications usually included in APEX installation

†where '/d' means a digit in the range [0,9]

### File name format:

<database name>\_f<application id>.sql

## 4.3 Files Export

The tool built exports all the files from APEX\_WORKSPACE\_FILES view for version 4 and APEX\_WORKSPACE\_STATIC\_FILES (see section 4.4 for more detail) view for version 5. All the files are exported in their original format (referred to as raw) and as SQL import scripts. The SQL script format provides an easy way to restore the files from backup, as the script can be simply run and the file will be imported. The file itself in the script is represented as a hex binary string, procedures in the script set the workspace id and application id (0 - if it's a workspace file), also any file with same name and workspace are removed before the new file is inserted. Figure 5 shows a simple example, where the penguin image is represented in the SQL import script extracts.

Files with ".sql" and ".zip" file extensions were not included in the export, because these files are normally import files that have been saved after an import, even though all components were successfully imported and are included in the overall export.

### Files excluded from export:

- ✘ file extension: '.sql'
- ✘ file extension: '.zip'



and the file name. It was noticed, that in APEX 4 if you insert a file with the same name in the same workspace, an error is gotten, but the file is still inserted. Meanwhile all backups are stored with workspace id, file id, file name, so that it is easier for tracking.

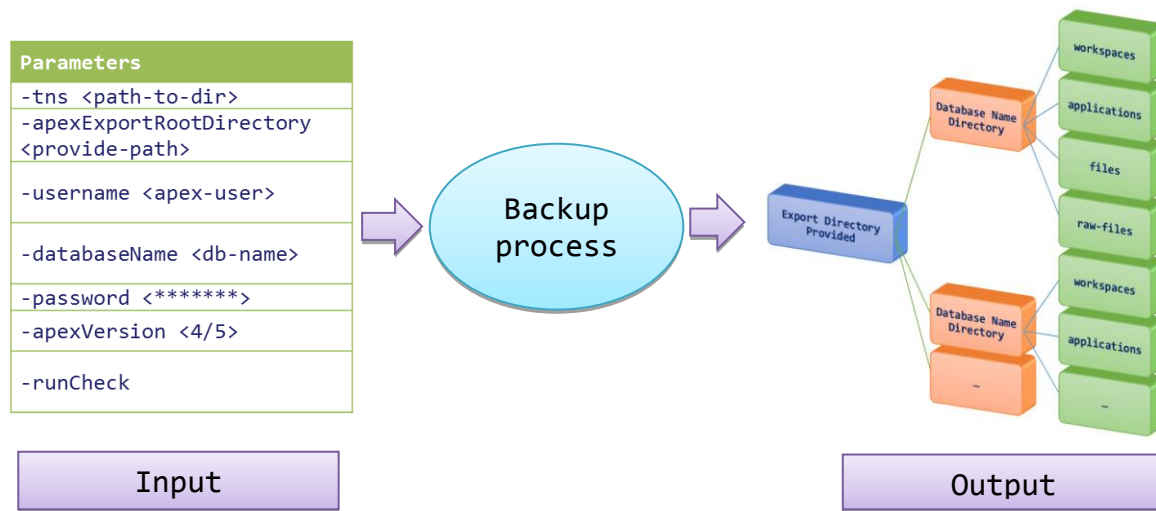


Figure 6. Diagram illustrating the big picture of the backup process.

## 4.5 Running Instructions

Running the tool built is very simple, all the code and executables are available at:

<https://gitlab.cern.ch/db/apex-backup-tool>

The tool can be run using two methods a java jar (executable) or from code. In both cases it is required to run the export as the APEX user (username usually starts with: 'APEX\_0'), otherwise not everything would be included. Here are the two methods to run the tool:

1) `java -jar apex-backup-tool.jar <params>`

OR

2) The Java project it can be run from `main.RunCERNEExporter.main(<params>)`. (The libraries from the lib folder need to be included in the java build path in order to run it successfully).

The tool takes in a number of parameters as input they are all thoroughly described in Table 3. They include a path to where the files should be output, database connection details, apex version and an option whether the test suite needs to be run. All the exported files are exported in the hierarchical directory structure described in Section 4.

Parameter	Description
-tns<path-to-dir>	TNS file parent directory.
-apexExportRootDirectory <provide-path>	Path to the parent directory where you want all the exports stored.
-username <apex-user>	APEX username to connect to the database. (Important to run with APEX user for full export!)
-databaseName<db-name>	Name of db that you want to connect to. DB name must be present in TNS and is case sensitive
-password <*****>	Password to the database
-apexVersion<4/5>	Integer version of apex, for now apex 4 and 5 supported.
-runCheck	Inclusion of this parameter, defines if you want to run and log an automatic check to verify if all files were exported

*Table 3.Explanation of parameters that need to be passed into the tool for successful backup.*

While the tool is run all export actions are logged and all the checks are also logged, thus all errors should be easily traceable. Also, the runtime is the last log entry of the tool.

## 5 Evaluation of Backup Tool

### 5.1 Automated Testing

A test suite is included in the tool in order to automate testing after each backup. It was an important part of the tool, because as soon as one tries to export a larger APEX installation it is very hard to keep track of all files.

The test suite checks if files for all expected workspaces and applications were exported. The same check is carried out for the expected workspace static files. The check is done by checking the workspace / application / file id. For raw files it is also checked if the export was successful by comparing the MD5 hash of the file blob in the workspace to the MD5 of the file exported.

Moreover, all the plug-in files, application static files and theme files are also checked to see if they are included in the export by parsing the application exports for included file names. No more checks are carried out for these files, because they are included by the official Oracle export utility. This way all files are accounted for.

In the case when there is a mismatch between expected and exported files an "InstanceExportException" is thrown with a meaningful message. All exports and checks are logged so that errors can be easily traceable.

### 5.2 User Testing

Further testing was carried out by users to see if anything is missing and if all the restored applications are functioning as expected. As currently all important applications are in APEX 4, two workspaces with complex applications, they were exported from APEX 4 and imported to APEX 5 to a database that contained copied schemas. Then the procedure was repeated on APEX

5, the imported workspaces on 5 were exported and imported to a different APEX 5 installation. This was done because testing backing up APEX 5 applications was a priority and there were no complex applications on such an installation.

All restored versions in APEX 4 and 5 were examined with cooperation of LHC configuration team and IT-DB developers, who looked at the more complex applications with their expertise knowledge. They did not report any errors concerning APEX.

### 5.3 Testing on a database

Lastly, the tool was used to back up a development database with 64 workspaces, 356 applications and 5450 files. It took around 12 minutes to carry out the task without any database disruption. This is the largest APEX (4) installation at CERN. Similarly to the testing in section 5.2 the APEX 4 backup was restored in APEX 5, backed up again, and restored on another installation of APEX 5.

The backed up and imported components were checked by looking at the APEX schemas. The backups were successful all the expected files were included. The import from version 4 to 5, showed that only one file of the same name per workspace was imported, as expected by the logic (see section 4.4), however to ensure that the right one was kept, the issue in APEX 4 needs to be fixed. When importing from versions 5 to 5 all the expected files were included. Moreover, when importing the larger files (of file formats such as ".mp4", ".exe"... ) the database was running out of PGA memory. This has to be taken into account when carrying out imports.

## 6 Importing the APEX Backups

The last step of the project was writing a script that would import all the backed up applications to a completely new installation of APEX. The order in which all the scripts are run is essential for a successful restore of the whole APEX environment. First the workspaces have to be exported, then the applications and lastly the external files. This is because of the nested structure that was presented in Figure 2 of Section 2.

### 6.1 Import Wrapper

A simple solution was written as a bash wrapper that runs all the PL/SQL scripts on the database, given the database connection details and the parent folder of the database back-up. It iteratively imports workspaces, applications and files. Within each import a connection to the database is made, to ensure that one broken import (due to size or other) would not kill the ones following.

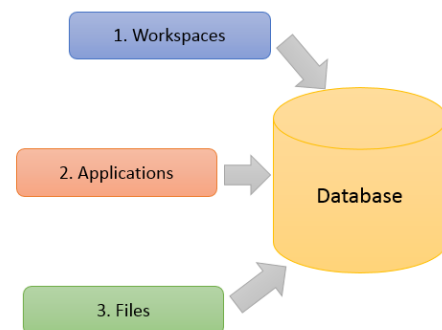


Figure 1.

### 6.2 Running Instructions

Just as with the backup tool running the import wrapper built is very simple, all the code and executables are available at: <https://gitlab.cern.ch/db/apex-backup-tool>. The bash script is under the import tool directory. The script has to be run on the machine where the database is

stored and with the connection details of the APEX user. The import can be done using the following command (see table 4 for parameter explanation):

```
bash import-script.sh <parent-directory-of-export><db-username/db-password>
```

---

Parameter	Description
<parent-directory-of-export>	Path to the directory named according to the database exported.
<db-username/db-password>	APEX user connection details (username that starts with "APEX_0" and password.)

---

*Table 4. Import wrapper parameter explanation.*

## 7 Conclusion

All in all, two tools were built one for backing up the APEX environment and one to enable restoring it. This was done as the evaluation step of the available backup solutions showed that a lot of curtail functionality is missing.

The Java back up tool can perform online backups from APEX versions 4 and 5 that include necessary the APEX components and necessary external files of all types and sizes. The tool has a test suite included that ensures that all expected components were indeed exported. The tool was tested beyond automated testing, complex applications were backed up, restored and examined by experienced APEX developers to see if anything is missing. It was tried out on the largest APEX installation on a development database at CERN and the backup went through successfully.

In the case of data corruption, a new APEX installation will be installed on the database. The wrapper import tool will be used to restore the exported backups. Successful restore is guaranteed by iterating through and importing all the files in the correct order to restore the APEX environment.

After a little more testing with other kinds of applications the tool should be ready for production. All the code and executables are available on CERN's git repository.



## 8 References

- [1] Application Express Architecture, <http://www.oracle.com/technetwork/developer-tools/apex/apex-arch-086399.html>
- [2] Oracle Application Express Getting Started, <http://www.oracle.com/technetwork/developer-tools/apex/application-express/apex-getting-started-1863613.html>
- [3] Tim Fox, John Scott, and Scott Spendolini. 2011. *Pro Oracle Application Express 4* (2nd ed.). Apress, Berkely, CA, USA.
- [4] APEX 5: Deprecated and Desupported Features and Changed Behavior by Marko Grociki, <http://apexbyg.blogspot.ch/2015/04/apex-5-deprecated-and-desupported.html>