API and Database Versioning Compatibility Script - HOW TO GUIDE

# FOLDER STRUCTURE FOR TEMPLATING

Following is the folder structure for the Template structure to be adhered to in order for the script to be run successfully.

folderName

version-execute-sql-shell-script.sh (Script to be Run)

PREVIOUS\_VERSION-to-NEXT\_VERSION (Version Upgrade Script)

1 – SampleScript1.sql

2 – SampleScript2.sql

3 – SampleScript3.sql

sequential-execute.txt

NEXT\_VERSION-to-PREVIOUS\_VERSION (Rollback Script)

1 – OtherSampleScript1.sql

2 – OtherSampleScript2.sql

3 – OtherSampleScript3.sql

sequential-execute.txt

Both the Version Upgrade and Rollback is a necessary pre-requisite for each new versioning change. Also care must be taken to make the naming format of the folder to adhere the convention of ‘{PREVIOUS\_VERSION}-to-{NEXT\_VERSION}’

The file sequential-execute contains a list of all the SQL scripts to be executed in order. For example the file sequential-execute.txt will contain the following three lines delimited by the newline character to execute it in order (All files must end with a carriage return to the new line):

1-SampleScript1.sql

2-SampleScript2.sql

3-SampleScript3.sql

# VERSIONING

There are primarily two styles of semantic versioning to be adhered to:

1. The first one is the numerical versioning standard intended for releases:

{MAJOR\_VERSION}.{MINOR\_VERSION}.{PATCH}

* **Major:** Major updates are non-compatible, meaning consumers can not upgrade without changing their software where applicable.
* **Minor:**  Minor updates are backward compatible new releases added to the software
* **Patch**: Bug fix, Performance improvement, environment or internal tweaks

NOTE: In the context of our Database Versioning, we will be ensuring compatibility up till the Minor Version Level meaning that both Major and Minor versions must be matched in order for the API to be compliant with a said version.

1. The second one is a textual representation of newly added features will be represented by an underscore-separated name for the feature.

new\_feature\_name

For example, the two possible naming conventions can arise are:

* “2.1.2-to-2.2.0” *and* “2.2.0-to-2.1.2”
* “2.1.2-to-new\_feature” *and* “new\_feature-to-2.1.2”

# USAGE

Following is the usage of the shell script



* **Folder Name:** Name of the folder which must be run (For example: 2.1.2-to-2.2.0 *and* 2.2.0-to-2.1.2 )
* **Profile:** The profile whose settings must be applied.
* **Comment:** The (Optional) comment to be added regarding the execution for this shell script

NOTE: If the credentials for the database connection must be changed then look for the following lines in code and edit them accordingly.

