

# Backup Up Essbase Cloud (EssCS) Applications

The good news is migrating to the cloud doesn't change a lot when it comes to backing up your Essbase applications. Conceptually, it is the same. The utilities used are slightly different.

## Enter CLI

If you are new to Essbase on the cloud, the CLI, or command line interface, is something you will want to download and configure. It is a pretty useful utility and easy to use. I will say that it is new and missing a lot of functionality you may want. I was just as frustrated using EPMAutomate with PBCS when it came out. Three years later, however, EPMAutomate is pretty complete. I am hoping for the same progression with CLI. For backing up your apps, the CLI will give you everything you need.

## Running An LCM Backup

There is really only one command that is a must. I will get into why I say this in a second. LcmExport will run an LCM and store it locally, which is a nice bonus. There is no need for any other commands to download and rename it.

LcmExport has the following parameters.

- -verbose (or -v) will provide a more complete response description, especially if there is an error
- -application (or -a) requires an additional parameter that identifies your application name
- -localdirectory (or -ld) requires an additional parameter that tell the command where to store the backup file

- `-zipfilename` (or `-z`) requires an additional parameter and is the name of the LCM file that everything will be stored locally
- `-threads` (or `-T`) requires an additional parameter equal to the number of threads you want to use to run the backup
- `-skipdata` (or `-skip`) will tell the LCM to ignore the data in the application
- `-overwrite` (or `-o`) is used will tell the process to overwrite the zip file if it exists
- `-password` (or `-p`) requires an additional parameter to send the password to the command

At minimum, the application and zipfilename parameters are required. The localdirectory and overwrite parameters will likely be used in every call you make as well.

```
C:\[cli foldername]\esscs.bat lcmExport -a Sample -z Sample.zip -ld c:/temp -o
```

## Why This Isn't Enough

I have always felt very strongly that data exports should be done because corruption will remain in the pag files until it is fixed, and often times, it isn't found for days, weeks, or months. At that point, you can't export your data and you are in real trouble. So, I don't rely solely on the LCM backup strategy.

There are a couple ways to export the data. With an on-premise implementation you might use Maxl to export the data. The other option is to write a calculation that does the exports. The calculations route will provide more option with formatting, the delimiter, and what data is included. It might be a little slower, but since the inclusion of this option, I have relied on it ever since.

You could integrate Maxl at this point to do the same thing, but the CLI also provides you with the tools to do it if you

use a calculation. At this point, assume a calculation exists named FullExport that exports the data to application and database path on the server with a name of FullExport.txt.

At this point, there are two additional commands that will bulletproof your backup strategy.

```
C:\[cli foldername]\esscs calc -a Sample -d Basic -s FullExport.csc
```

```
C:\[cli foldername]\esscs download -f FullExport.txt -a Sample -d Basic -ld c:/backup -o
```

## Completing The Circle

Normally this would be executed through DOS, or my favorite, PowerShell. This would allow the dynamic generation of the scripts so they could be reused. Things like the application name, database name, local path, calc script, and possibly some others, would all be variables. The result would be something produced similar to this.

```
C:\cli_utility\esscs login -url https://myEssbase-test-myDomain.analytics.us2.oraclecloud.com/essbase -u kylegoodfriend
```

```
C:\cli_utility\esscs lcmExport -a Sample -z Sample.zip -ld c:/Backups -o
```

```
C:\cli_utility\esscs calc -a Sample -d Basic -s FullExport.csc
```

```
C:\cli_utility\esscs download -f FullBackup.txt -a Sample -d Basic -ld c:/Backups -o
```

I would add a step in my shell to rename this LCM and the data export downloaded with a date and time in the name.

```
REM Rename the LCM Zip file
```

```
ren c:\Backups\Sample.zip Sample_%date:~10,4%%date:~4,2%%date:~7,2%.zip
```

```
REM Rename the data export
```

```
ren c:\Backups\FullExport.txt Sample_Backup_%date:~10,4%%date:~4,2%%date:~7,2%.txt
```

```
REM Delete all files older than 30 days
```

```
forfiles /p c:\Backups /s /m *.* /D -30 /C "cmd /c del @path"
```

## A Final Note

The calculation script referenced above would look something like this. There are many options that can be set. If you aren't familiar with this, a quick google will get you what you need.

```
SET DATAEXPORTOPTIONS
{
  DataExportLevel "LEVEL0";
  DataExportDynamicCalc OFF;
  DataExportNonExistingBlocks OFF;
  DataExportRelationalFile ON;
  DataExportOverwriteFile ON;
}
DATAEXPORT "File" "," "/Sample/Basic/FullExport.txt";
```

As always, post and share. If you have a question, do the same.