

Create BMC Control-M usage reports from CLI instead GUI

Date: 2021-08-21
modified: 2021-08-21
tags: Control-M, Code, Java
description: Control-M usage reports from CLI
category: Code
slug: create-bmc-control-m-usage-reports-via-cli-instead-gui
Author: Dominik Wombacher
lang: en
transid: create-bmc-control-m-usage-reports-via-cli-instead-gui
Status: published

I had to solve a quite interesting problem recently, the Task was to generate Control-M usage reports on a monthly basis and send them via E-Mail.

Control-M provides a Solution for that: **BMC Control-M Usage Reporting Tool**

But it's a Java GUI Application that can't be automated or run in Batch mode. So I decided to write a small "wrapper" in Java which re-use the original Classes from the Control-M Usage Reporting Tool (Version 9.0.00).

That way the results are 100% identical without any (unsupported) customizing but it can be executed from the command line.

I want to make clear: All Control-M Copyrights belong to BMC, including the Java Code and Classes. All what my little wrapper does, is loading the Class which comes with the original Tool and call the same methods as it would be done in the GUI to generate the report.

Therefore I want to share the Code of my wrapper and the necessary cmd file (yes, all this is only running on Windows due to hardcoded backslashes in file paths in the used Class) to start it.

Downloading the required Jar file from BMC, Building the Project (I used "Export: Runnable JAR file" in Eclipse), adding the necessary *data/env.txt* and *data/config.ini* is up to you. I suggest you take a look into the directory structure and files of the *BMC Control-M Usage Reporting Tool*, it isn't that hard figure it out and get it working :)

ctmusagereport

```
package ctmusagereport;
import java.io.File;
import com.bmc.ctmem.usagetool.Enhancement.ZipFiles;
import com.bmc.ctmem.usagetool.dbaccess.AddOnsReport;
import com.bmc.ctmem.usagetool.dbaccess.DBAccess;
import com.bmc.ctmem.usagetool.dbaccess.DBAccessManager;
import com.bmc.ctmem.usagetool.dbaccess.EndpointReport;
import com.bmc.ctmem.usagetool.dbaccess.JobCountReport;
import com.bmc.ctmem.usagetool.dbaccess.Params;
public class Main {
    public static void main(String[] args) {
        // Used to Load Env and Config File also to Generate and Get Report Directory
        Params par = Params.instance();
        try {
            par.LoadEMEnvironmentsFromFile();
```

```
        } catch (Exception e) {
            e.printStackTrace();
```

```

    }
    try {
        par.LoadConfigParamFromFile();
    } catch (Exception e) {
        e.printStackTrace();
    }

    // Initiates DB Connection based on values in data/env.txt
    DBAccessManager dbam = DBAccessManager.instance();
    dbam.RefreshDBAccessManager();

    // Get DB Connection for Environment 0 (first line in env file)
    DBAccess con = dbam.GetDBAccess(0);
    con.GetEMVersionFromDB();

    // Creates new Directory in reports/ with datetime now ("yyyy-MM-dd-HH:mm:ss")
    par.CreateReportDir();

    // Trigger Generation of all Reports identical to GUI (Wizard / FlowManager)
    new JobCountReport(par.GetReportDirName());
    new AddOnsReport(par.GetReportDirName());
    new EndpointReport(par.GetReportDirName(), true);

    // For easier handling, add all generated files to "reports.zip" in Generated Report Directory
    File[] fileReportsDir = new File(par.GetReportDirName()).listFiles();
    ZipFiles.Zip(fileReportsDir, par.GetReportDirName());
}
}
}

```

ctmusagereport.cmd

```
java -cp ".*;ctmusagereport_lib/*" ctmusagereport.Main
```

I tested it successfully with "OpenJDK11U-jdk_x64_windows_hotspot_11.0.12_7" from <https://adoptium.net/> on Windows Server 2016.

What I like about working with Control-M is that you can Script and Customize almost everything somehow, even worse that I had to find a workaround like that for the usage reporting, would be great if BMC would offer such an CLI option with the original Tool.