# DB Upgrade Automation

This document describes the solution for automation of  database upgrade process.
Document also covers new database installation process.

Blagoj Kostoski (blagoj.kostovski@navetti.com)
2016-04-26
Navetti AB
Phone: +46 8 44 00 120

# Contents

# 1 DB Upgrade automation

## 1.1 Background

NPP4 is modern ERP solution which is used to propagate changes (price calculations) on large amounts of data in near-to-real-time time period.

The product is build as compact solution which is highly configurable in order to meet each client specific requirements.

Each client has it's specific environment, setup, background processes, data size and diversity.

All these constraints makes each client a specific pattern on top of which NPP4 is performing.

Every PricePoint version upgrade consist of database upgrade and application upgrade. Database upgrade is performed manually by executing several sql scripts containing all changes that needs to be performed. Sometimes manual intervention in the scripts is needed in order to perform the upgrade.

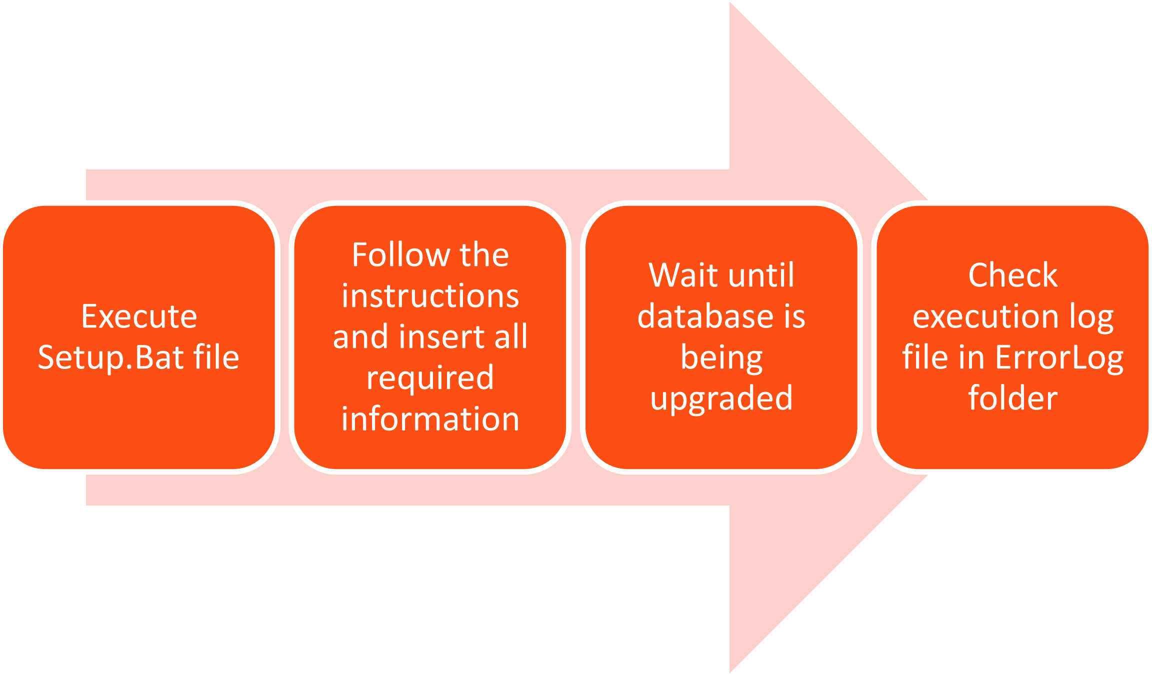
The idea is to automate the process of database upgrade installation as much as we can. Automation of new database installation is also covered with this document.

## 1.2 Solution

We have created a solution wich uses single BATCH script for managing the upgrade process.

The BATCH script executes all sql scripts on the target server and store the output of the execution in text file. If any error occurs the upgrade process will be rolled back entirely.

User should execute Setup.bat file and follow the instructions. This is simple diagram for actions that needs to be performed by end user:

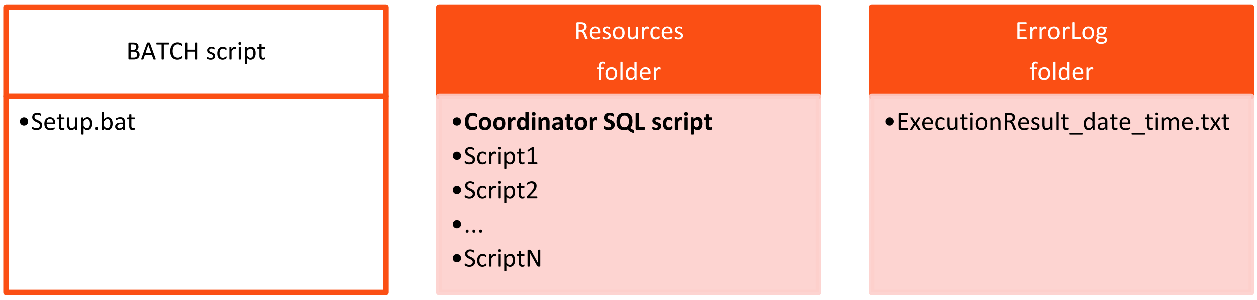

Execute Setup.Bat file
Follow the instructions and insert all required information
Wait until database is being upgraded
Check execution log file in ErrorLog folder

# 2 Solution Description

## 2.1 Structure

The package consists of:

1. BATCH script (Setup.BAT)
2. Coordinator SQL script (00_Coordinator.SQL)
3. Resources folder, with N sql scripts (\path\Resources)
4. ErrorLog folder (\path\ErrorLog)



BATCH script

- Setup.bat

Resources folder

- **Coordinator SQL script**
- Script1
- Script2
- ...
- ScriptN

ErrorLog folder

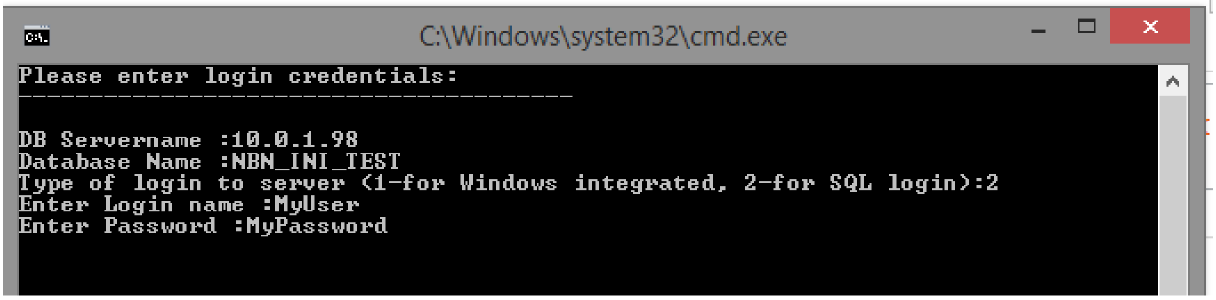
- ExecutionResult_date_time.txt

Diagram of folder structure for the solution

### 2.1.1 BATCH script

Script is named as **Setup.bat** and it is the only file needed to be executed for the upgrade. Setup requires information about target server and user credentials:

- DB Server name
- Database name
- Type of authentication
- Login/Password if SQL login selected



```
Please enter login credentials:
---------------------------------------

DB Servername :10.0.1.98
Database Name :NBN_INI_TEST
Type of login to server (1-for Windows integrated, 2-for SQL login):2
Enter Login name :MyUser
Enter Password :MyPassword
```

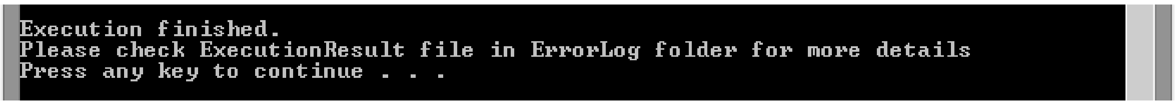
Picture 1: Setup.bat execution input parameters

If additional information needs to used for the upgrade process, then the batch script needs to be extended to read this information in variable.

Example for this kind of information can be Name of the file group for table creation, name of some custom object on client database or anything that can be client specific information and requires user intervention.

When all input information will be inserted Setup.bat executes Coordinator SQL script using inserted credentials and passes all input data to the coordinator script.

Execution finishes with informational message **Execution finished.**



```
Execution finished.
Please check ExecutionResult file in ErrorLog folder for more details
Press any key to continue . . .
```

Picture 2: Setup.bat execution end

### 2.1.2 Coordinator SQL script

Coordinator script is placed in the Resources subfolder and by default it is named as **00_Coordinator.sql**. The name can be changed if needed, but setup.bat file also needs to be changed to use this script. It can be done by changing the granted value to StartingFile variable with new file name.

set StartingFile=00_Coordinator.sql -> set StartingFile=NewCoordinatorName.sql

Coordinator script contains execution for all files from the upgrade package. It also contains logic for error handling and ROLLBACK functionality in case of errors.

Printing the file currently executing is good practice, since in case of error only the error line is returned which is not sufficient information to locate the error.

Coordinator script contains the following code:

```
SET NOCOUNT ON
GO

SET XACT_ABORT ON
GO

    BEGIN TRANSACTION
     PRINT 'currently executing: 01Sprint1.sql'
    :r $(CurrentFolder)01Sprint1.sql

    PRINT 'currently executing: 02Sprint2.sql'
    :r $(CurrentFolder)02Sprint2.sql

    /*
      ....
      list all files from the upgrade package
    */
    GO

   IF (XACT_STATE()=1)
    BEGIN
      PRINT 'All scripts executed successfully'
      COMMIT TRANSACTION
    END
    ELSE IF (XACT_STATE()=-1)
    BEGIN
      PRINT 'Error in execution!!!'
      PRINT 'All changes has been rolled back. Please find error details below...'
      ROLLBACK TRANSACTION
    END

GO
```

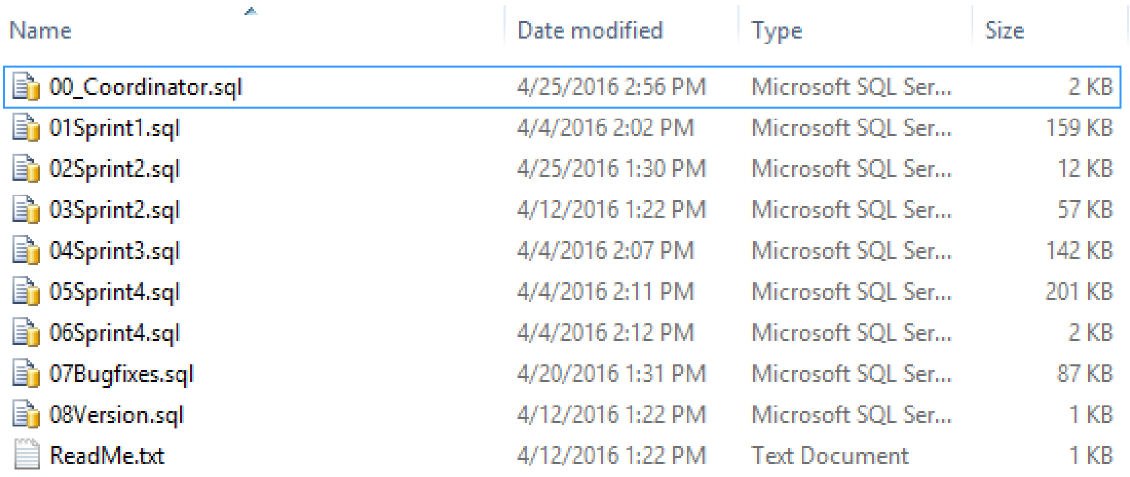
### 2.1.3 Resources folder

Except the Coordinator script, resources folder contains all SQL scripts needed for the database upgrade.

Although it is not a must, it is good practice this files to be named starting with number (01,02,..) followed by the name of the sprint or the name of the feature/functionality.

File names must be *without blank spaces* in order to be processed by coordinator script.

If any additional information needs to be passed to the upgrade scripts, it can be done by using variables written as `$(VariableName).`
Coordinator script will make simple string replacement and will execute the script using `:r` Command.



| Name | Date modified | Type | Size |
|---|---|---|---|
| 00_Coordinator.sql | 4/25/2016 2:56 PM | Microsoft SQL Ser... | 2 KB |
| 01Sprint1.sql | 4/4/2016 2:02 PM | Microsoft SQL Ser... | 159 KB |
| 02Sprint2.sql | 4/25/2016 1:30 PM | Microsoft SQL Ser... | 12 KB |
| 03Sprint2.sql | 4/12/2016 1:22 PM | Microsoft SQL Ser... | 57 KB |
| 04Sprint3.sql | 4/4/2016 2:07 PM | Microsoft SQL Ser... | 142 KB |
| 05Sprint4.sql | 4/4/2016 2:11 PM | Microsoft SQL Ser... | 201 KB |
| 06Sprint4.sql | 4/4/2016 2:12 PM | Microsoft SQL Ser... | 2 KB |
| 07Bugfixes.sql | 4/20/2016 1:31 PM | Microsoft SQL Ser... | 87 KB |
| 08Version.sql | 4/12/2016 1:22 PM | Microsoft SQL Ser... | 1 KB |
| ReadMe.txt | 4/12/2016 1:22 PM | Text Document | 1 KB |

Picture 3: Example structure of Resources folder

### 2.1.4 ErrorLog folder

ErrorLog folder contains separate file for every execution of the Setup.bat file. Files are named as ExecutionResult_YYYYMMDD_HHMMSS.log and this is simple output message from database version upgrade:

```
currently executing: 01Sprint1.sql
Script execution finished
currently executing: 02Sprint2.sql
Msg 2714, Level 16, State 6, Server DEV-PP-K8-02, Line 2
There is already an object named 'ImportType' in the database.
```

In case of error in some script, error message is saved in the output file, and whole upgrade is rolled back. Any print messages in the sql files are written in ExecutionResult file, and it is important this messages to be more descriptive.

## 2.2 Prerequisites

Batch script execution doesn’t require special permissions. However credentials used to connect to database needs to be sufficient for performing speficic actions.

For upgrade database at least db_ddladmin , db_datareaded and db_datawriter permissions are required. If upgrade contains agent job creation or  modification, user with sufficient permissions on SQL Server Agent is required.

For new database installation at least dbcreator server role is required.

## 2.3 Best practices

Although solution contains Rollback functionality in case of error, database backup before applying the upgrade is recommended.