

Ins & Outs of sp_execute_external_script - I

A drilldown into sp_execute_external_script

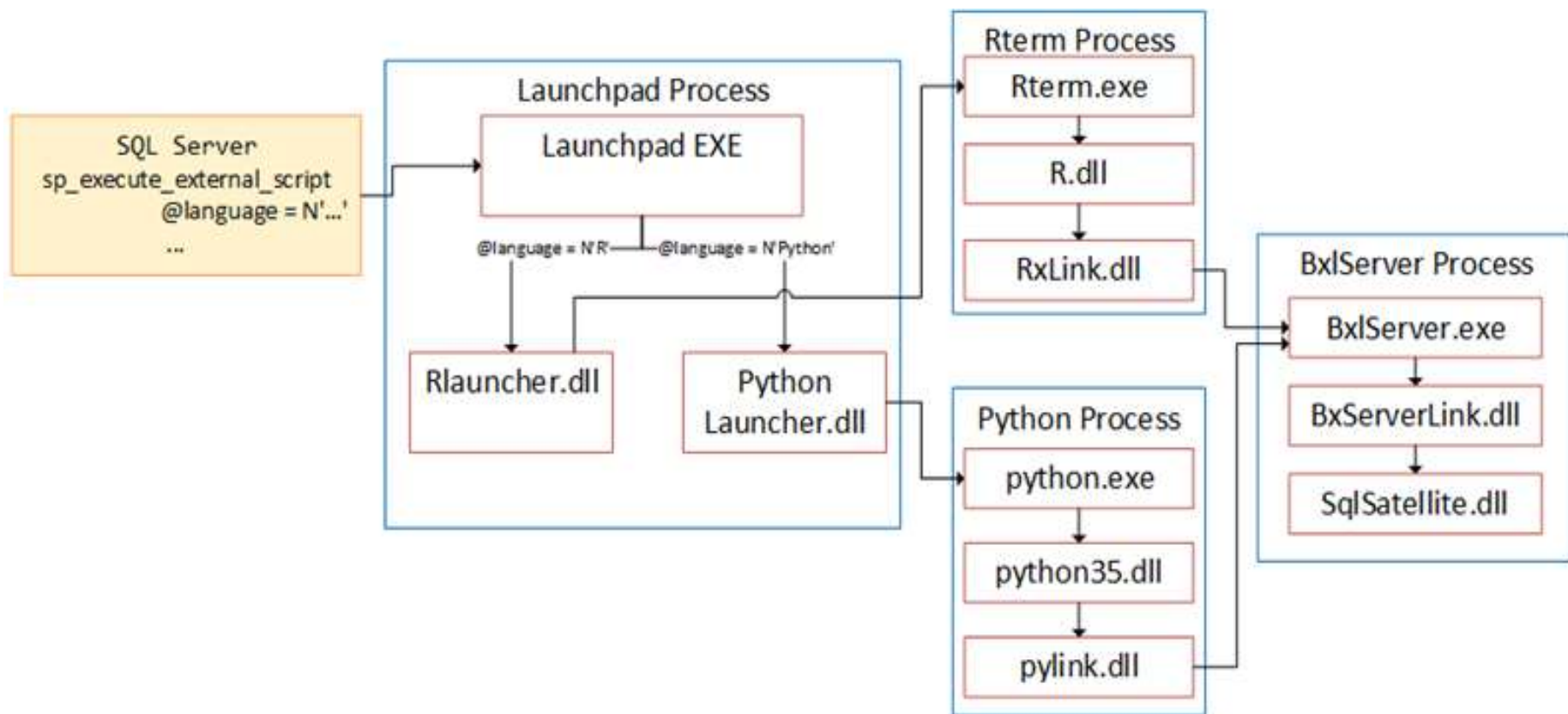
Agenda

- SPEES (sp_execute_external_script)
- Permissions
- Parameters
 - @language
 - @script
 - @input_data_1
 - @input_data_1_name
 - @output_data_1_name

sp_execute_external_script (SPEES)

- Entry point into the external engine (R, Python -- for now)
- Similar to [sp_executesql](#).
- Executes the script provided as argument, at an external location.
- External scripts need to be enabled
 - SQL Server instance needs to be restarted
- Certain permissions needed

Architecture



Permissions

- Grant rights to user.
- User needs right to execute SPEES (unless dbo).
 - beware SPEES is system proc - grant in master.
- User needs EXECUTE EXTERNAL SCRIPT (unless dbo).

```
-- grant execution  
USE master  
GRANT EXECUTE ON sp_execute_external_script to public;  
GO
```

```
-- grant rights on scripts  
GRANT EXECUTE ANY EXTERNAL SCRIPT TO user1;  
GO
```

SPEES Signature

```
sp_execute_external_script
    @language = N'language',
    @script = N'script',
    [ @input_data_1 = 'input_data_1' ]
    [ , @input_data_1_name = N'input_data_1_name' ]
    [ , @output_data_1_name = N'output_data_1_name' ]
    [ , @parallel = 0 | 1 ]
    [ , @params = N'@parameter_name data_type [ OUT | OUTPUT ] [ ,...n ]' ]
    [ , @parameter1 = 'value1' [ OUT | OUTPUT ] [ ,...n ] ]
[WITH <execute_option>]
```

Parameters - I

- `@language`: Indicates what external engine to use.
- `@script`: The script to execute. Can be literal as well as variable as well as script-file.
 - Read in the script from a table.

Parameter @script

```
EXEC sp_execute_external_script @language = N'R',  
    @script = N'iris_dataset <- iris  
    setosa <- iris[iris$Species == "setosa",]  
    meanSepWidth <- mean(setosa$Sepal.Width)  
    cat(paste("Seposa sepal mean width: ", meanSepWidth))';
```

```
DECLARE @scriptParam nvarchar(max) = N'iris_dataset <- iris  
    setosa <- iris[iris$Species == "setosa",]  
    meanSepWidth <- mean(setosa$Sepal.Width)  
    cat(paste("Seposa sepal mean width: ", meanSepWidth))';
```

```
EXEC sp_execute_external_script @language = N'R',  
    @script = @scriptParam;
```

```
EXEC sp_execute_external_script @language = N'R',  
    @script = N'source("c:/rscripts/iris_r.r")';
```


Parameters - II

- `@input_data_1`: Statement to generate dataset to push in to the external engine.
 - No, there is no `@input_data_2`, etc.
 - Only one dataset can be pushed into the engine.
 - You can use `UNION ALL` / retrieve data in the script
- `@input_data_1_name`: Name to use in script for the dataset pushed in (defaults to `InputDataSet`).

Input Data - I

```
EXEC sp_execute_external_script @language = N'R',  
    @script = N'iris_dataset <- InputDataSet  
    setosa <- iris_dataset[iris_dataset$Species == "setosa",]  
    meanSepWidth <- mean(setosa$SepalWidth)  
    cat(paste("Seposa sepal mean width: ", meanSepWidth))',  
    @input_data_1 = N'SELECT * FROM dbo.tb_irisdata_full'
```

```
EXEC sp_execute_external_script @language = N'R',  
    @script = N'iris_dataset <- InputDataSet  
    setosa <- iris_dataset[iris_dataset$Species == "setosa",]  
    meanSepWidth <- mean(setosa$SepalWidth)  
    cat(paste("Seposa sepal mean width: ", meanSepWidth))',  
    @input_data_1 = N'SELECT * FROM dbo.tb_irisdata_even UNION ALL  
    SELECT * FROM dbo.tb_irisdata_uneven';
```

Input Data - II

```
EXEC sp_execute_external_script @language = N'R',
    @script = N'# set up connection string
    sqlConnString <- "Driver=SQL Server;server=win10-dev;
                    database=IrisTestDb;uid=<some_uid>;pwd=<some_pwd>"

    # define the data
    mydata <- RxSqlServerData(table = NULL, sqlQuery = "SELECT *
                FROM dbo.tb_irisdata_uneven", connectionString = sqlConnString)

    # open the dataset
    rxOpen(mydata)
    # read the data
    iris_uneven <- rxReadNext(mydata)
    versicolor <- iris_uneven[iris_uneven$Species == "versicolor",]
    meanSepWidthVersi <- mean(versicolor$SepalWidth)
    # get the data from the input data set
    iris_dataset <- InputDataSet
    setosa <- iris_dataset[iris_dataset$Species == "setosa",]
    meanSepWidth <- mean(setosa$SepalWidth)

    # output the data
    cat(paste("Seposa sepal mean width:", meanSepWidth, ".",
              "Versicolor sepal mean width:", meanSepWidthVersi)),
    @input_data_1 = N'SELECT * FROM dbo.tb_irisdata_even';
```

<http://nielsberglund.com>

Parameters - III

- `@output_data_1_name`: Name to use for dataset returned (default is `OutputDataSet`).
- `WITH` - used to define output data set.
 - not a param per se, and not tied to SPEES.

Output Data

```
EXEC sp_execute_external_script
  @language = N'R',
  @script = N'
    iris_dataset <- MyDataSet
    setosa <- iris_dataset[iris_dataset$Species == "setosa",]
    meanSepWidth <- mean(setosa$SepalWidth)
    cat(paste("Seposa sepal mean width: ", meanSepWidth))
    multiplier <- 5
    iris_dataset$SepalLength <- iris_dataset$SepalLength * multiplier
    SepalLengthMult <- data.frame(iris_dataset$SepalLength)',
  @input_data_1 = N'SELECT * FROM dbo.tb_irisdata_full',
  @input_data_1_name = N'MyDataSet',
  @output_data_1_name = N'SepalLengthMult'
WITH RESULT SETS ((LengthMultiplied float));
```

Summary

- `sp_execute_external_script` entrypoint to external engine
- If not dbo, permissions required
- Parameters to define language and code to execute.
- Parameters to define input and output dataset.