

LAB #11

Server Automation. Managing Steps

Managing Steps

After creating the task, we must create its steps, for which the `sp_add_jobstep` stored procedure is used. Its syntax:

```
sp_add_jobstep
[ @job_name = ] 'job_name'
, [ @step_name = ] 'step_name'
, [ @step_id = ] 'step_id'
[ , [ @subsystem = ] 'subsystem' ]
[ , [ @command = ] 'step_code' ]
[ , [ @database_name = ] 'database_name' ]
[ , [ @database_user_name = ] 'database_user_name' ]
[ , [ @retry_attempts = ] retry_attempts ]
[ , [ @retry_interval = ] retry_interval ]
```

Consider arguments.

- **'job_name'**. The name of the job for which the step is created.
- **'step_name'**. The name of the step.
- **'step_id'**. The identifier of the step.
- **'subsystem'**. Is used by the SQL Server Agent service for execute the step code defined in the **'step_code'** argument. The default value is 'TSQL' (Transact_SQL Language).
- **'step_code'**. The code to be executed on this step.
- **'database_name'**. The database name, for which are executed the statements and stored procedures. This argument should only be specified if the **@subsystem** argument is specified as 'TSQL'.
- **'database_user_name'**. The name of database user, who executes the step. This argument should only be specified if the **@subsystem** argument is specified as 'TSQL'.
- **retry_attempts**. The number of attempts to perform this step.
- **retry_interval**. Used with the previous argument. Allows you to specify the time interval (in minutes) between attempts to repeat the step.

Examples.

Create the Step_51 step. The identifier of the step is 1. Add the step to the Job_51 job. The subsystem is 'TSQL'. The code to be executed on this step is 'SELECT * FROM Customer;'. Specify 51 retry attempt, with each retry to occur after 51 minute wait. The database user User_51 executes the step. For the database 'Ordering' is executed the statement.

USE master;

CREATE LOGIN Login_51 WITH PASSWORD = 'password51'; --If Login_51 not exists

USE Ordering;

CREATE USER User_51 FOR LOGIN Login_51; --If User_51 not exists

USE msdb;

```
EXEC dbo.sp_add_job @job_name = N'Job_51';           --If Job_51 not exists
EXEC sp_add_jobstep
@job_name = N'Job_51',
@step_name = N'Step_51',
@step_id = 1,
@subsystem = 'TSQL',
@command = 'SELECT * FROM Customer;',
@retry_attempts = 51,
@retry_interval = 51,
@database_name = N'Ordering',
@database_user_name = N'User_51';
```

Create the Step_52 step. The identifier of the step is 1. Add the step to the Job_52 job. The subsystem is 'TSQL'. The code to be executed on this step is 'BACKUP DATABASE Ordering TO Ordering_Backup;'. Specify 52 retry attempts, with each retry to occur after a 52 minutes wait. The database user User_52 executes the step. For the database 'Ordering' is executed the statement.

```
USE master;
CREATE LOGIN Login_52 WITH PASSWORD = 'password52'; --If Login_52 not exists
USE Ordering;
CREATE USER User_52 FOR LOGIN Login_52;           --If User_52 not exists
USE msdb;
EXEC dbo.sp_add_job @job_name = N'Job_52';       --If Job_52 not exists
EXEC sp_add_jobstep
@job_name = N'Job_52',
@step_name = N'Step_52',
@step_id = 1,
@subsystem = 'TSQL',
@command = 'BACKUP DATABASE Ordering TO Ordering_Backup;',
@retry_attempts = 52,
@retry_interval = 52,
@database_name = N'Ordering',
@database_user_name = N'User_52';
```

Create the Step_53 step. The identifier of the step is 1. Add the step to the Job_53 job. The subsystem is 'TSQL'. The code to be executed on this step is 'ALTER INDEX ALL ON Customer REBUILD;'. Specify 53 retry attempts, with each retry to occur after a 53 minutes wait. The database user User_53 executes the step. For the database 'Ordering' is executed the statement.

```
CREATE LOGIN Login_53 WITH PASSWORD = 'password53'; --If Login_53 not exists
USE Ordering;
CREATE USER User_53 FOR LOGIN Login_53;           --If User_53 not exists
```

```

USE msdb;
EXEC dbo.sp_add_job @job_name = N'Job_53';           --If Job_53 not exists
EXEC sp_add_jobstep
@job_name = N'Job_53',
@step_name = N'Step_53',
@step_id = 1,
@subsystem = 'TSQL',
@command = 'ALTER INDEX ALL ON Customer REBUILD;',
@retry_attempts = 53,
@retry_interval = 53,
@database_name = N'Ordering',
@database_user_name = N'User_53';

```

Create the Step_54 step. The identifier of the step is 1. Add the step to the Job_54 job. The subsystem is 'TSQL'. The code to be executed on this step is 'UPDATE Contract SET paid_money_Lari = 2500 WHERE contract_ID = 54;'. Specify 54 retry attempts, with each retry to occur after a 54 minutes wait. The database user User_54 executes the step. For the database 'Ordering' is executed the statement.

```

CREATE LOGIN Login_54 WITH PASSWORD = 'password54'; --If Login_54 not exists
USE Ordering;
CREATE USER User_54 FOR LOGIN Login_54;           --If User_54 not exists
USE msdb;
EXEC dbo.sp_add_job @job_name = N'Job_54';           --If Job_54 not exists
EXEC sp_add_jobstep
@job_name = N'Job_54',
@step_name = N'Step_54',
@step_id = 1,
@subsystem = 'TSQL',
@command = 'UPDATE Contract SET paid_money_Lari = 2500 WHERE contract_ID = 54;',
@retry_attempts = 54,
@retry_interval = 54,
@database_name = N'Ordering',
@database_user_name = N'User_54';

```

Create the Step_55 step. The identifier of the step is 1. Add the step to the Job_55 job. The subsystem is 'TSQL'. The code to be executed on this step is 'DELETE FROM Staff WHERE staff_ID = 55;'. Specify 55 retry attempts, with each retry to occur after a 55 minutes wait. The database user User_55 executes the step. For the database 'Ordering' is executed the statement.

```

CREATE LOGIN Login_55 WITH PASSWORD = 'password55'; --If Login_55 not exists
USE Ordering;
CREATE USER User_55 FOR LOGIN Login_55;           --If User_55 not exists

```

```

USE msdb;
EXEC dbo.sp_add_job @job_name = N'Job_55'; --If Job_55 not exists
EXEC sp_add_jobstep
@job_name = N'Job_55',
@step_name = N'Step_55',
@step_id = 1,
@subsystem = 'TSQL',
@command = 'DELETE FROM Staff WHERE staff_ID = 55;',
@retry_attempts = 55,
@retry_interval = 55,
@database_name = N'Ordering',
@database_user_name = N'User_55';

```

Change the step Step_51 of Job_51. The new code to be executed on this step is 'SELECT * FROM Constract WHERE debt_Dolari >= 150;'. Specify 15 retry attempts, with each retry to occur after a 15 minutes wait. The database user User_15 executes the step:

```

CREATE LOGIN Login_15 WITH PASSWORD = 'password15'; --If Login_15 not exists
USE Ordering;
CREATE USER User_15 FOR LOGIN Login_15; --If User_15 not exists
USE msdb;
EXEC sp_update_jobstep
@job_name = N'Job_51',
@step_name = N'Step_51',
@step_id = 1,
@subsystem = 'TSQL',
@command = 'SELECT * FROM Constract WHERE debt_Dolari >= 150;',
@retry_attempts = 15,
@retry_interval = 15,
@database_name = N'Ordering',
@database_user_name = N'User_15';

```

Change the Step_52 step. The new code to be executed on this step is 'SELECT * INTO Customer_20 FROM Customer;'. Specify 25 retry attempts, with each retry to occur after a 25 minutes wait. The database user User_25 executes the step:

```

CREATE LOGIN Login_25 WITH PASSWORD = 'password25'; --If Login_25 not exists
USE Ordering;
CREATE USER User_25 FOR LOGIN Login_25; --If User_25 not exists
USE msdb;
EXEC sp_update_jobstep
@job_name = N'Job_52',
@step_name = N'Step_52',
@step_id = 1,

```

```

@subsystem = 'TSQL',
@command = 'SELECT * INTO Contract_20 FROM Customer;',
@retry_attempts = 25,
@retry_interval = 25,
@database_name = N'Ordering',
@database_user_name = N'User_25';

```

Delete the Step_52 step in the Job_52 job. The step identifier is 1:

```

USE msdb;
EXEC sp_delete_jobstep
        @job_name = N'Job_52',
        @step_id = 1;

```

Association of job with server

For association of job with server is used the **sp_add_jobserver** stored procedure. It's syntax:

```

sp_add_jobserver [ @job_name = ] 'job_name'
                [ , [ @server_name = ] 'server_name' ]

```

Associate the Job_51 job with the Server_51 local server.

```

USE msdb ;
EXEC sp_add_jobserver
        @job_name = N'Job_51',
        @server_name = N'GTU-PCN\SQL_2012_2';

```

Exercises.

1. Create the Step_1 step in the Job_1 job, which executes command „SELECT * FROM Staff;“ and defines 2 attempts every 2 minute. The subsystem is ‘TSQL’. The step identifier is 1.
2. Create the Step_2 step in the Job_1 job, which opens the Baza_1 database. Owner of job is the U_1 user. The subsystem is ‘TSQL’. The step identifier is 1.
3. Change the database, user and command, that is executed by the Step_3 of the Job_1 job. The new command - „SELECT * FROM Staff;“. New database is Ordering. New user is U_2. Tis step defines 2 attempts after every 2 minutes. The step identifier is 1.
4. Delete the Step_2 step from the Job_1 job. The step identifier is 1.
5. Associate the Job_1 job with the Server_1 local server.