

# MSSQL

- [Обслуживание 1C MSSQL](#)
- [MSSQL борщ](#)

# Обслуживание 1С MSSQL

## Материалы по вопросу

<https://plast.com.kz/config-sql-server-1c-maintenance-plans/>

<https://its.1c.ru/db/metod8dev#content:5837:hdoc:p4>

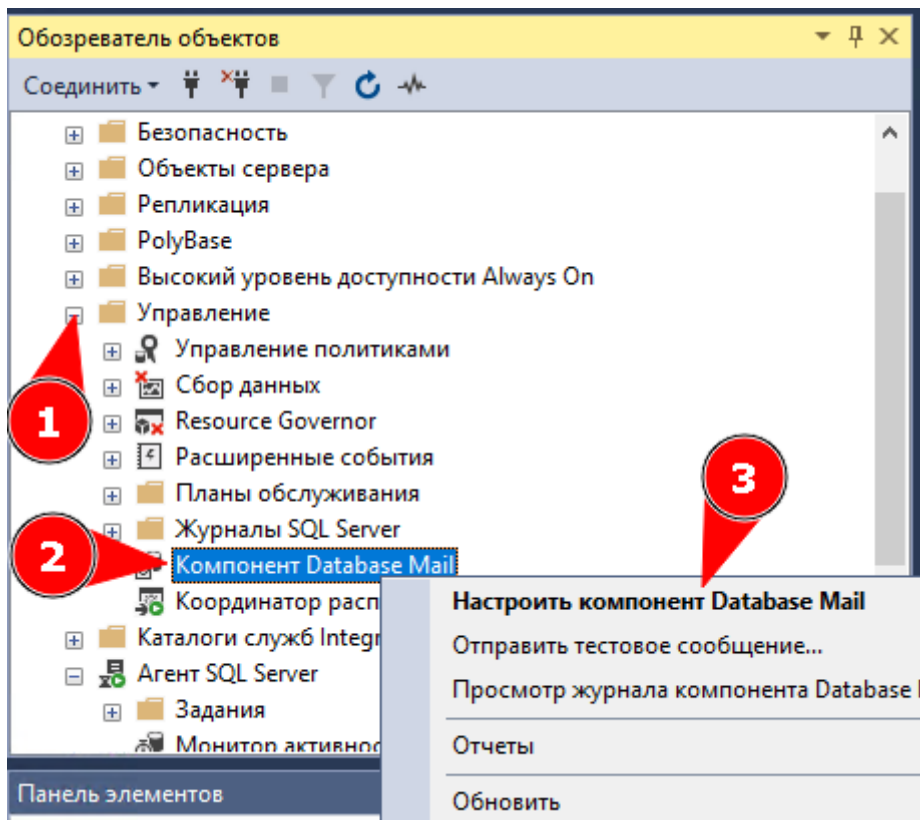
[https://interface31.ru/tech\\_it/2012/08/obslyzhivanie-baz-1s-v-ms-sql-server-chast-2.html](https://interface31.ru/tech_it/2012/08/obslyzhivanie-baz-1s-v-ms-sql-server-chast-2.html)

[https://interface31.ru/tech\\_it/2012/02/obslyzhivanie-baz-1s-v-ms-sql-server-chast-1.html](https://interface31.ru/tech_it/2012/02/obslyzhivanie-baz-1s-v-ms-sql-server-chast-1.html)

<https://forum.infostart.ru/forum86/topic289736/>

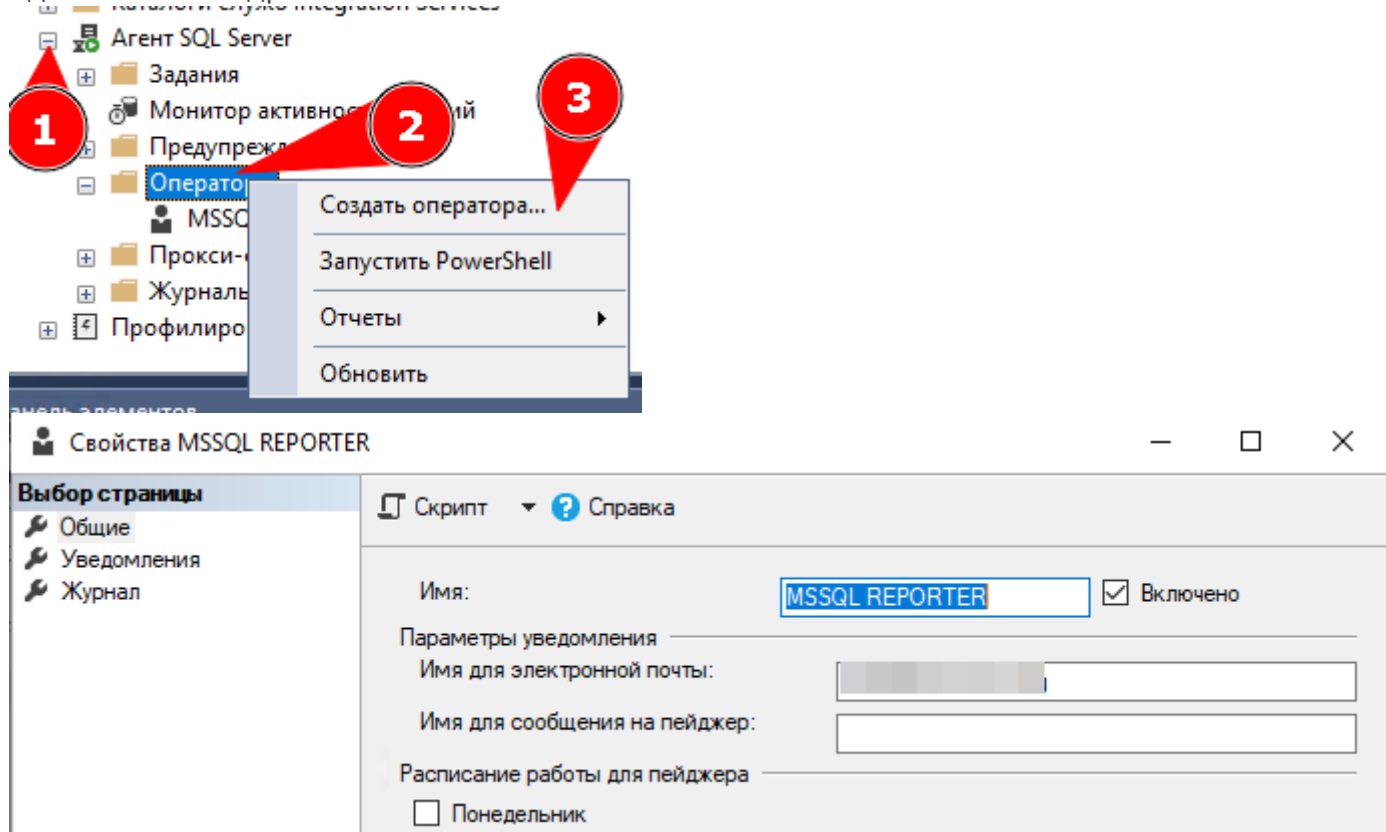
<http://www.gilev.ru/forum/viewtopic.php?f=15&t=1844>

## Email уведомления

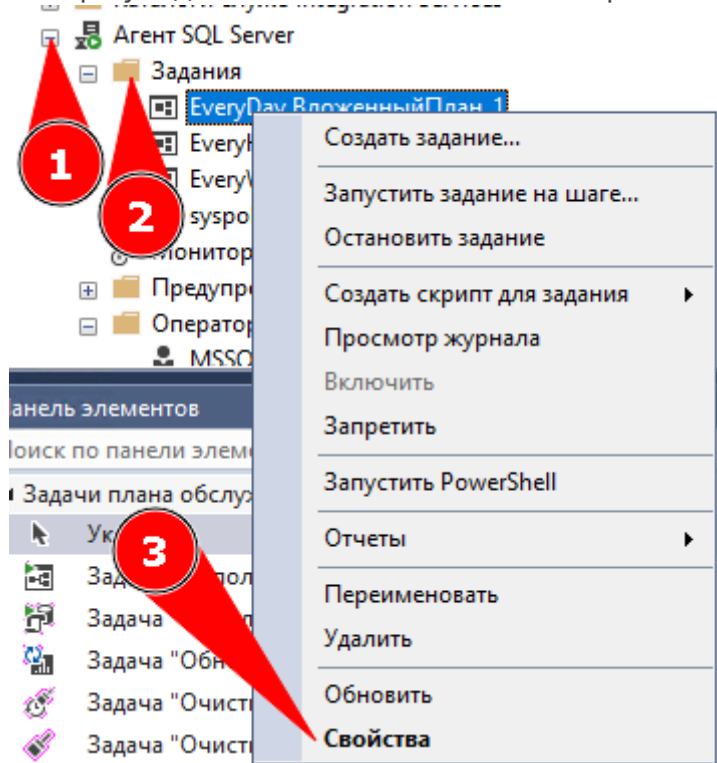


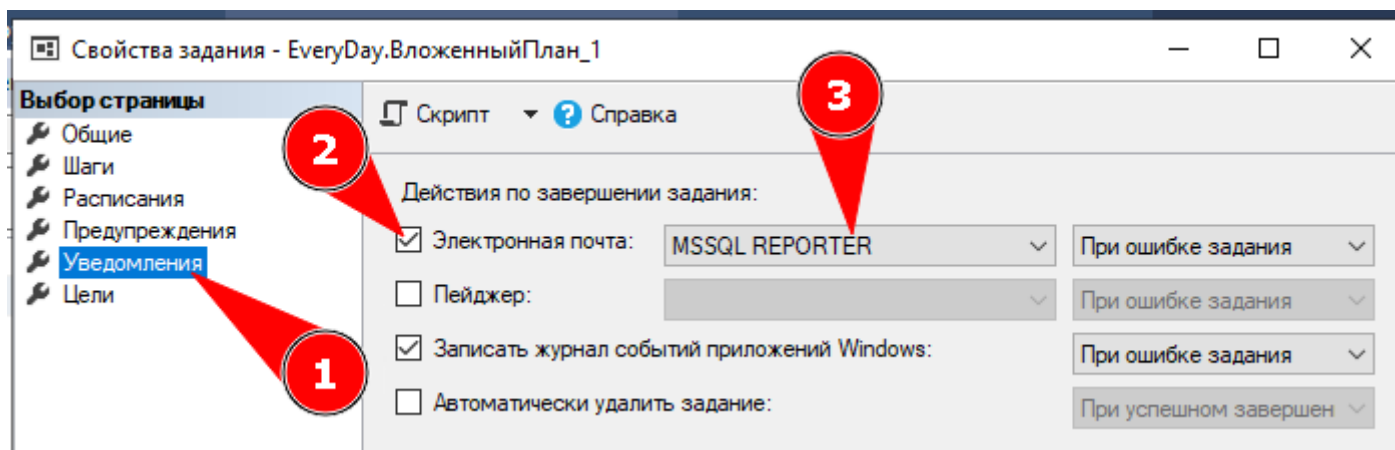
Нужно ввести параметры почты **от имени кого** будут отправлены уведомления.

Заведите оператора, укажите адрес **куда** надо направлять уведомления. В моём случае это один и тот же адрес.

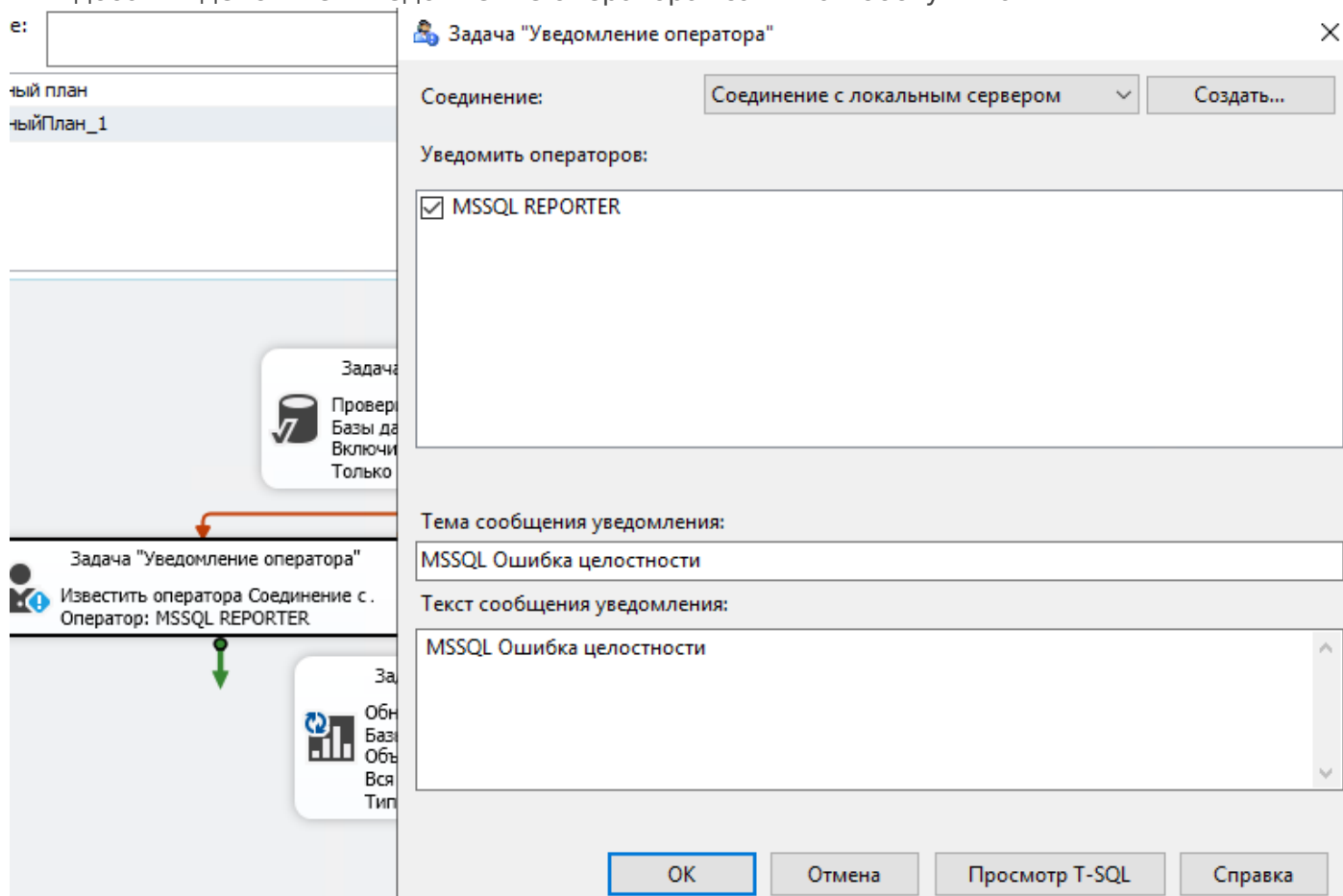


Теперь уведомление можно включить при ошибке выполнения плана обслуживания:





Или добавив действие "Уведомление оператора" сам план обслуживания:



# MSSQL борщ

## Включить индексирование таблиц 8.3.22

Реорганизация индексов вызывала ошибку по причине: **поскольку отключена блокировка на уровне страницы**

```
USE [basename]
GO

DECLARE @object_schema VARCHAR(256);
DECLARE @object_name VARCHAR(256);
DECLARE @index_name VARCHAR(256);

DECLARE db_cursor CURSOR FOR
SELECT OBJECT_SCHEMA_NAME(object_id) AS object_schema,
       OBJECT_NAME(object_id)      AS object_name,
       name                        AS index_name
FROM   sys.indexes
WHERE  allow_page_locks = 0 AND OBJECT_SCHEMA_NAME(object_id) != 'sys';

OPEN db_cursor;
FETCH NEXT FROM db_cursor INTO @object_schema, @object_name, @index_name;
WHILE @@FETCH_STATUS = 0
BEGIN

    EXEC ('ALTER INDEX ' + @index_name + ' ON ' + @object_schema + '.' + @object_name + ' SET
(ALLOW_PAGE_LOCKS = ON)');

    FETCH NEXT FROM db_cursor INTO @object_schema, @object_name, @index_name;

END;
CLOSE db_cursor;
DEALLOCATE db_cursor;
```

Взято [отсюда](#)

# Проверить фрагментацию MSSQL

```
DECLARE @db_id SMALLINT;

SET @db_id = DB_ID(N'MyBaseSQL');

IF @db_id IS NULL
BEGIN;
    PRINT N'Неправильное имя базы';
END;

ELSE
BEGIN;
    SELECT
        object_id AS [ID объекта],
        index_id AS [ID индекса],
        index_type_desc AS [Тип индекса],
        avg_fragmentation_in_percent AS [Фрагментация в %]
    FROM sys.dm_db_index_physical_stats(@db_id, NULL, NULL, NULL , 'LIMITED')
    ORDER BY [avg_fragmentation_in_percent] DESC;
END;

GO
```

Взято [отсюда](#)

# Остановить резервное копирование MSSQL

В SQL запроснике найти PID резервного копирования:

```
SELECT session_id as SPID, command, a.text AS Query, start_time, percent_complete,  
dateadd(second,estimated_completion_time/1000, getdate()) as estimated_completion_time  
FROM sys.dm_exec_requests r CROSS APPLY sys.dm_exec_sql_text(r.sql_handle) a  
WHERE r.command in ('BACKUP DATABASE','RESTORE DATABASE')
```

И завершить процесс (тут же, в запроснике):

KILL НомерПроцесса

# Шифрованные копии MSSQL

## Зашифровать копии

**Создайте учетные данные SQL Server.** Для создания учетных данных SQL Server подключитесь к ядру СУБД, откройте новое окно запроса, скопируйте в него следующий пример и нажмите кнопку **Выполнить**. (Я этот шаг пропустил)

```
CREATE CREDENTIAL mycredential  
WITH IDENTITY= 'mystorageaccount' - this is the name of the storage account you specified when creating a  
storage account  
, SECRET = '<storage account access key>' - this should be either the Primary or Secondary Access Key for the  
storage account
```

**Создайте главный ключ базы данных.** Выберите пароль для шифрования копии главного ключа базы данных, которая будет храниться в базе данных. Подключитесь к ядру СУБД, откройте новое окно запроса, скопируйте в него следующий пример и нажмите кнопку **Выполнить**. (Устанавливаете мастер ключ шифрования)

```
-- Creates a database master key.  
-- The key is encrypted using the password "<master key password>"  
USE Master;  
GO  
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<master key password>';  
GO
```

**Создайте сертификат резервной копии.** Создайте сертификат резервной копии в базе данных master. Вставьте следующий пример в окно запроса и нажмите **Выполнить**.

```
USE Master;
GO
CREATE CERTIFICATE MyTestDBBackupEncryptCert
    WITH SUBJECT = 'MyTestDBBackupEncryptCert ';
GO
```

**Выполните резервное копирование базы данных.** Укажите алгоритм шифрования и сертификат для использования. Скопируйте следующий пример в окно запроса и нажмите кнопку **Выполнить**.

```
BACKUP DATABASE [MyTestDB]
TO URL = N'C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Backup\MyTestDB.bak'
WITH
    CREDENTIAL 'mycredential' - this is the name of the credential created in the first step.
    ,COMPRESSION
    ,ENCRYPTION
    (
        ALGORITHM = AES_256,
        SERVER CERTIFICATE = MyTestDBBackupEncryptCert
    ),
    STATS = 10
GO
```

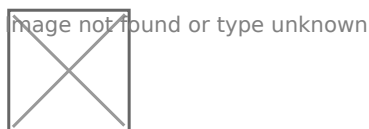
## Расшифровать копии

Взято [отсюда](#)

При попытке прочитать содержимое зашифрованной копии будет выдана ошибка “Cannot find server certificate with thumbprint”

```
Msg 33111, Level 16, State 3, Line 1
Cannot find server certificate with thumbprint '0xE11A199C1059C6F1E0223B56581CDCF3F043DFE8'.
Msg 3013, Level 16, State 1, Line 1
RESTORE DATABASE is terminating abnormally.
```

Сертификаты хранятся тут:



**Выгрузите информацию со старого сервера вот таким запросом.**



```
USE Master
go
BACKUP CERTIFICATE DB_Encrypt_Cert
TO FILE = 'Z:\Backup\DB_Encrypt_Cert.cer'
WITH PRIVATE KEY(
FILE = 'Z:\Backup\DB_Encrypt_Cert.prvk',
ENCRYPTION BY PASSWORD = 'StrongPassword'
)
```

### **Загрузите в новый вот таким:**

```
CREATE CERTIFICATE DB_Encrypt_Cert
FROM FILE = 'E:\MSSQL\DB_Encrypt_Cert.cer'
WITH PRIVATE KEY(
FILE = 'E:\MSSQL\DB_Encrypt_Cert.prvk',
DECRYPTION BY PASSWORD = '7Hx81GbNaxHP65rsSfiKAaVvKvN5beUY'
)
```

Теперь можно восстанавливать базу на новом сервере как обычно.