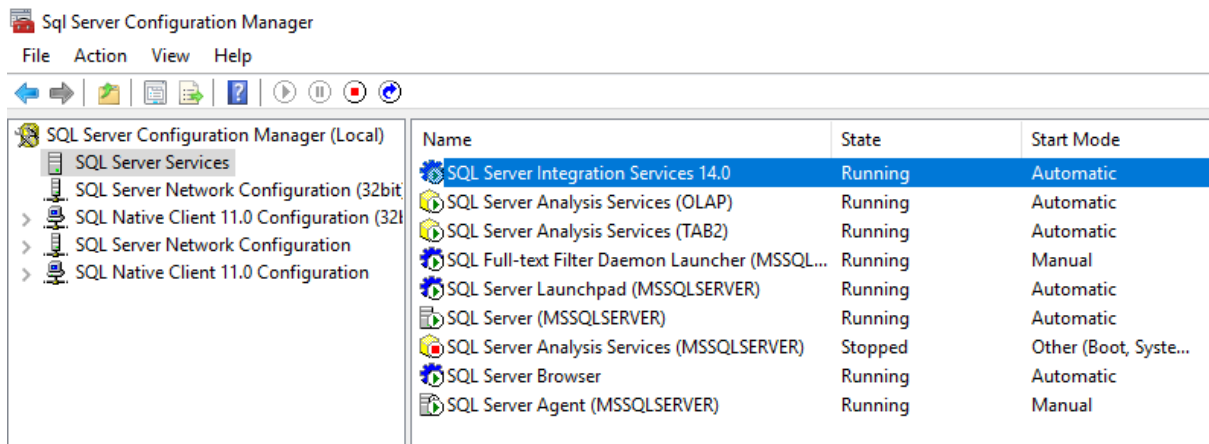


# Deployment, Konfiguration & Betrieb von SSIS

Prerequisites sind

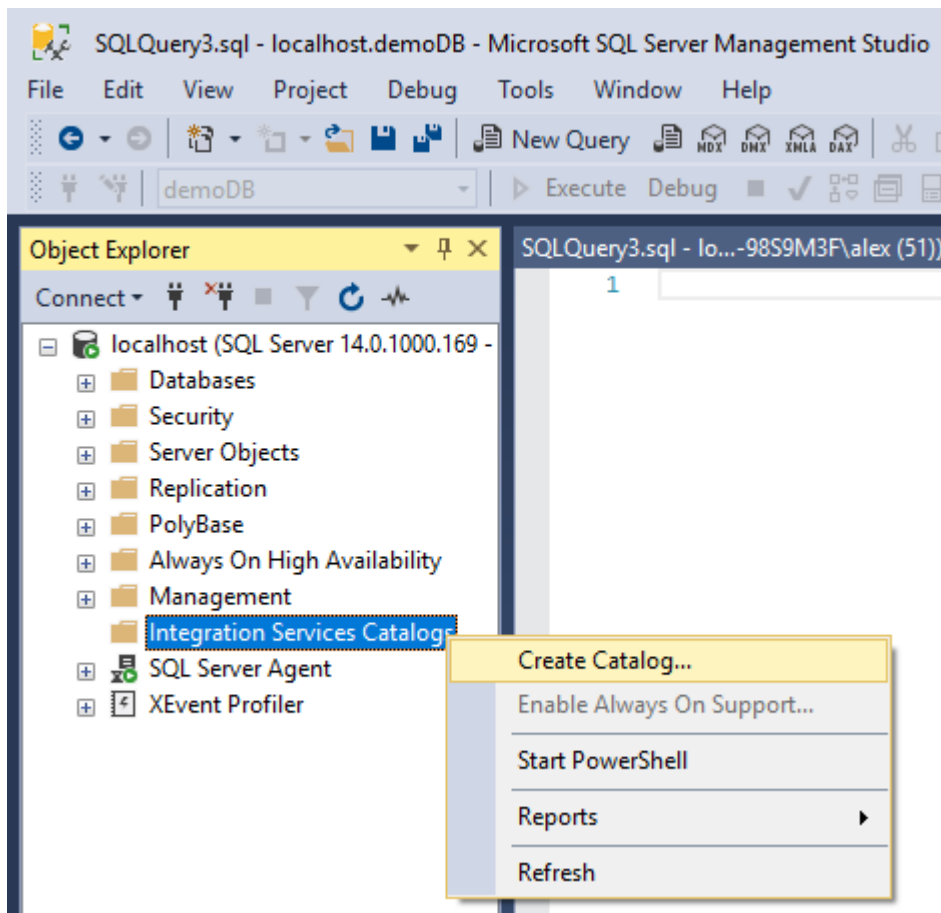
.. die installierten **SQL Server Integration Services**



Der **Sql Server Configuration Manager** ist in der MMC (Microsoft Management Console) inkludiert und kann auch via Commandline aufgerufen werden.

```
C:\Windows\SysWOW64\mmc.exe /32 C:\Windows\SysWOW64\SQLServerManager14.msc
```

Jetzt erfolgt das Erstellen einer „System“-Datenbank für die Integration Services.



Microsoft schreibt ..

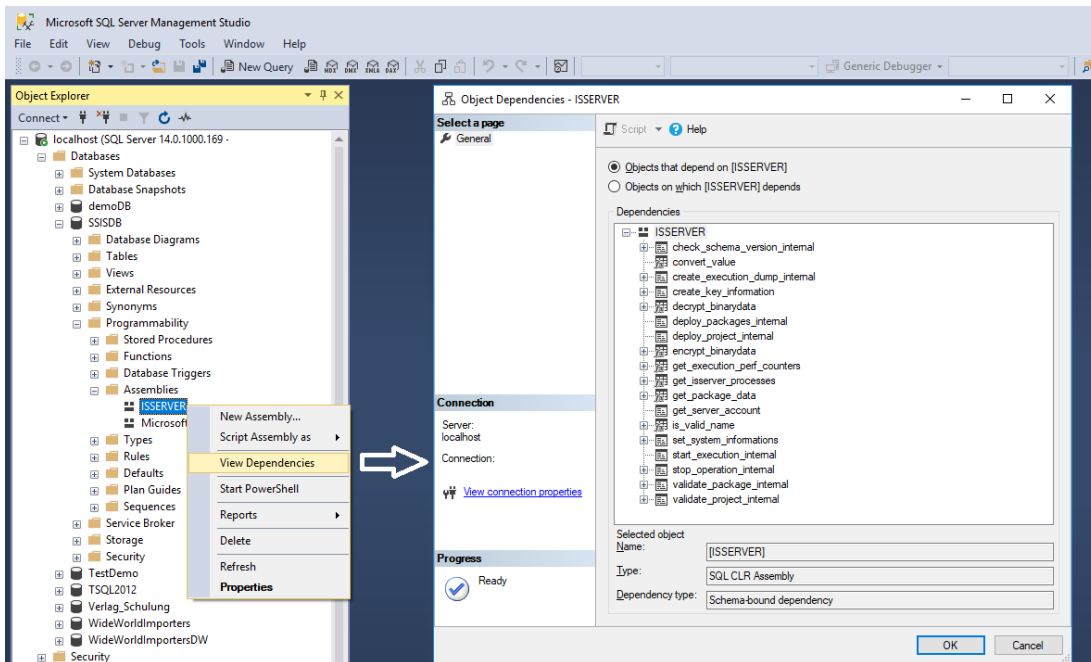
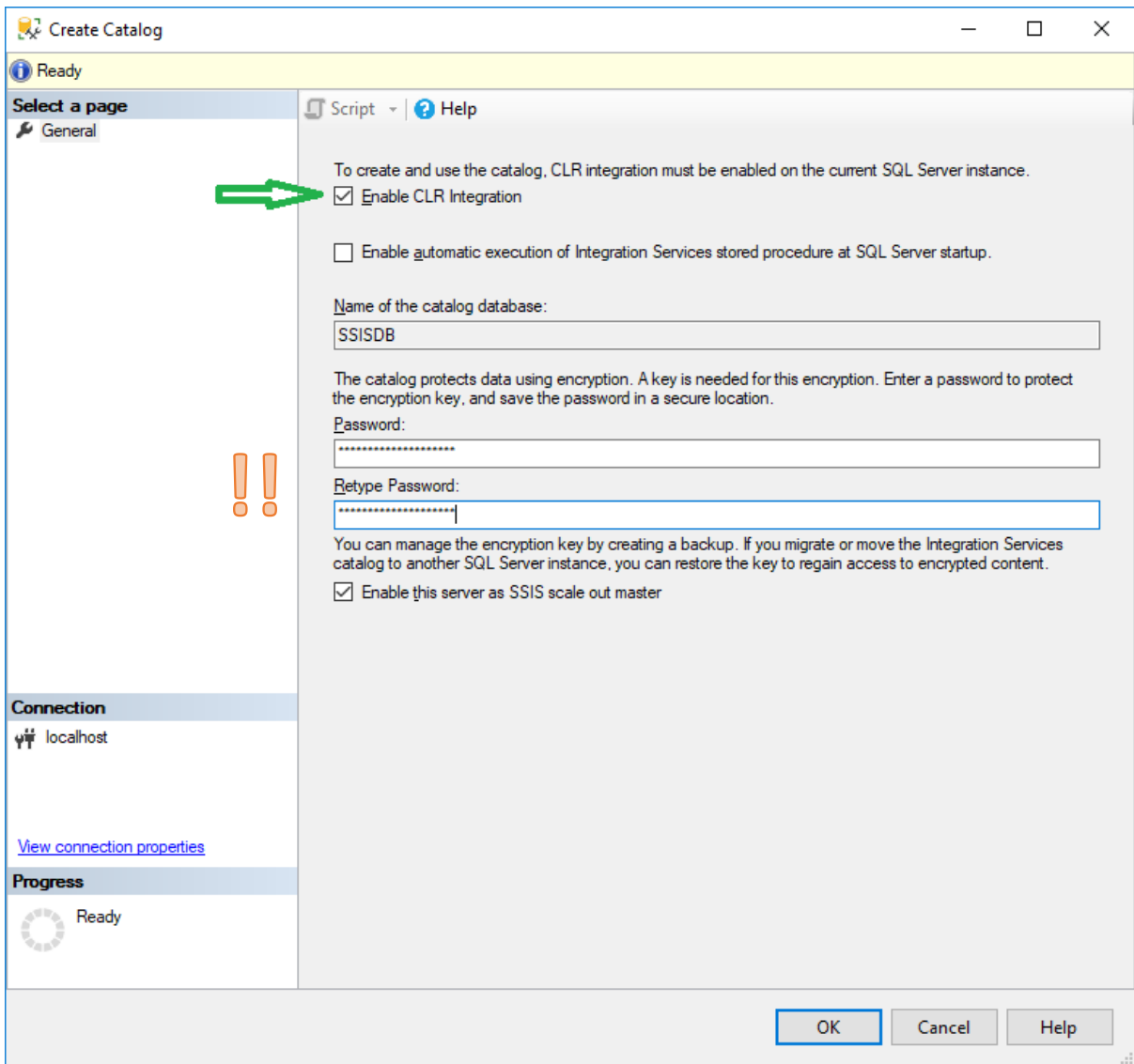
Der **SSISDB**-Katalog ist der zentrale Punkt zum Arbeiten mit Integration Services-Projekten (SSIS), die Sie auf dem Integration Services-Server bereitgestellt haben. Sie legen beispielsweise Projekt- und Paketparameter fest, konfigurieren Umgebungen, um Laufzeitwerte für Pakete anzugeben, führen Pakete aus, behandeln Paketprobleme und verwalten Integration Services - Servervorgänge ...

<https://docs.microsoft.com/de-de/sql/integration-services/catalog/ssis-catalog?view=sql-server-2017>

Weiterhin ist die Aktivierung der CLR-Komponente (Common Language Runtime) erforderlich

<https://docs.microsoft.com/de-de/dotnet/framework/data/adonet/sql/introduction-to-sql-server-clr-integration>

Die folgende Anzeige der Abhängigkeiten zeigt die gesamte Anzahl der „CLR-Programmierungen“ innerhalb der SSISDB.



Insbesondere ist hier auch auf die Eingabe des Passworts hingewiesen.

Es handelt sich dabei um den **Datenbank-Hauptschlüssel**.



Dieser wird für die Verschlüsselung verwendet, wenn Parameter und Konfigurationen enthalten sind ggf. unternehmenssensitive Werte. Der Schlüssel wird erstellt, wenn Sie den Katalog erstellen.

SSISDB password required - G... X  
SSIS-Katalog | Microsoft Docs X  
Zugriffsteuerung für sensible... X

https://docs.microsoft.com/de-de/sql/integration-services/catalog/ssidb-catalog?view=sql-server-2017

### So sichern Sie die SSIS-Datenbank

1. Öffnen Sie SQL Server Management Studio, und stellen Sie eine Verbindung zu einer Instanz von SQL Server her.

2. Sichern Sie den Hauptschlüssel für die SSISDB-Datenbank, indem Sie die Anweisung BACKUP MASTER KEY (Transact-SQL) verwenden. Der Schlüssel wird in einer von Ihnen angegebenen Datei gespeichert. Verwenden Sie ein Kennwort, um den Hauptschlüssel in der Datei zu verschlüsseln.

Weitere Informationen zur Anweisung finden Sie unter [BACKUP MASTER KEY \(Transact-SQL\)](#).

Im folgenden Beispiel wird der Hauptschlüssel in die Datei `c:\temp\directory\RCTestInstKey` exportiert. Das `LS2Setup!`-Kennwort wird zum Verschlüsseln des Hauptschlüssels verwendet.

```
backup master key to file = 'c:\temp\directory\RCTestInstKey'
encryption by password = 'LS2Setup!'
```

3. Sichern Sie die SSISDB-Datenbank mit dem Dialogfeld **Datenbank sichern** in SQL Server Management Studio. Weitere Informationen finden Sie unter [Gewusst wie: Sichern einer Datenbank \(SQL Server Management Studio\)](#).

4. Gehen Sie wie folgt vor, um das CREATE LOGIN-Skript für `##MS_SSIServerCleanupJobLogin##` zu generieren. Weitere Informationen finden Sie unter [CREATE LOGIN \(Transact-SQL\)](#).

- Erweitern Sie in SQL Server Management Studio im Objekt-Explorer die Knoten **Sicherheit** und **Anmeldungen**.
- Klicken Sie mit der rechten Maustaste auf `##MS_SSIServerCleanupJobLogin##`, und klicken Sie dann auf **Skript für Anmeldenamen als > CREATE in > Neues Abfrage-Editor-Fenster**.

5. Wenn Sie die SSISDB-Datenbank auf einer SQL Server-Instanz wiederherstellen, auf der der SSISDB-Katalog nie erstellt wurde, generieren Sie das CREATE PROCEDURE-Skript für „sp\_ssis\_startup“ wie folgt. Weitere Informationen finden Sie unter [CREATE PROCEDURE \(Transact-SQL\)](#).

a. Erweitern Sie im Objekt-Explorer den Knoten **Datenbanken** und dann die Knoten **master > Programmierbarkeit >**

**In diesem Artikel**

- [Features und Funktionen](#)
- [Katalogobjektbezeichner](#)
- [Katalogkonfiguration](#)
- [Berechtigungen](#)
- [Ordner](#)
- [Projekte und Pakete](#)
- [Parameter](#)
- [Serverumgebungen, Servervariablen und Serverumgebungsverweise](#)
- [Ausführungen und Überprüfungen](#)
- [Erstellen des SSIS-Katalogs](#)
- [Katalogeigenschaften \(Dialogfeld\)](#)
- [Sichern, Wiederherstellen und Verschieben des SSIS-Katalogs](#)
- [Aktualisieren des SSIS-Katalogs \(SSISDB\)](#)
- [Always On für den SSIS-Katalog \(SSISDB\)](#)
- [Verwandte Inhalte](#)

Ist diese Seite hilfreich?  
Ja Nein

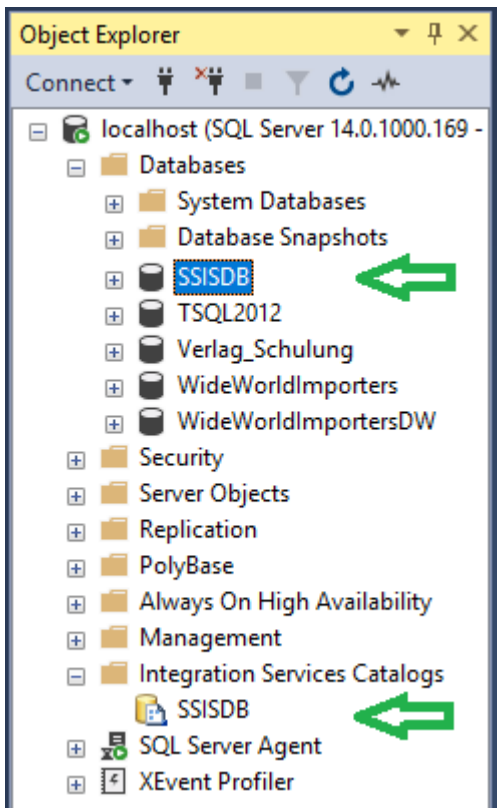
17:43  
08/07/2018

<https://docs.microsoft.com/de-de/sql/integration-services/catalog/ssidb-catalog?view=sql-server-2017>

Das Sichern des Datenbank-Hauptschlüssels, das Backup sowie das Datenbank-Restore ist hier

<https://docs.microsoft.com/de-de/sql/integration-services/catalog/ssidb-catalog?view=sql-server-2017>

ausführlich dokumentiert.



Ergebnis:

Die Datenbank „SSISDB“ ist erstellt; sie erscheint jedoch in der Auflistung der User-Datenbanken. Zusätzlich erscheint sie im Ordner Integration Services Catalogs.

Obleich die SSISDB als User-Datenbank erscheint ist es nicht empfehlenswert darin Schemaänderungen vorzunehmen. Weiterhin empfiehlt Microsoft nur Objecte aus dem Schema catalog zu verwenden.

Das Schema internal ist dagegen „undokumentiert“ und ausschließlich für Microsoft-Artefakte reserviert.

Für eigene Reporting-Zwecke kann z.B. das Schema „custom“ angelegt und verwendet werden. Dies ist u.A. im SSIS Catalog Dashboard von Andy Leonard realisiert.

Basic SSIS Catalog Dashboard in Power BI

https://dilmsuite.com/product/basic-ssis-catalog-dashboard-in-power-bi/

DILMSuite Contact Us Home Cart

Year	Month	Folder	Project	Package	Status
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success
2017	August	Report	Report	Report	Success

Project Total Execution Duration

Package Total Execution Duration

Home / DILMSuite / Basic SSIS Catalog Dashboard in Power BI

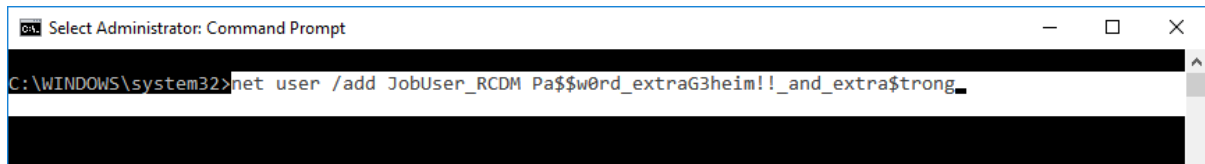
# Basic SSIS Catalog Dashboard in Power BI

**\$0.00**

1 Add to cart

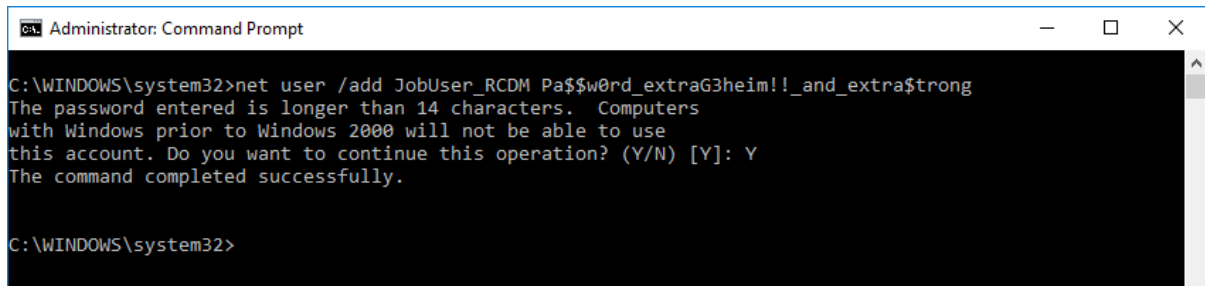
Category: DILMSuite

Für die weitere Konfiguration ist ein „technischer User“ empfehlenswert ( segregation of duties )



```
C:\WINDOWS\system32>net user /add JobUser_RCDM Pa$$w0rd_extraG3heim!!_and_extra$trong
```

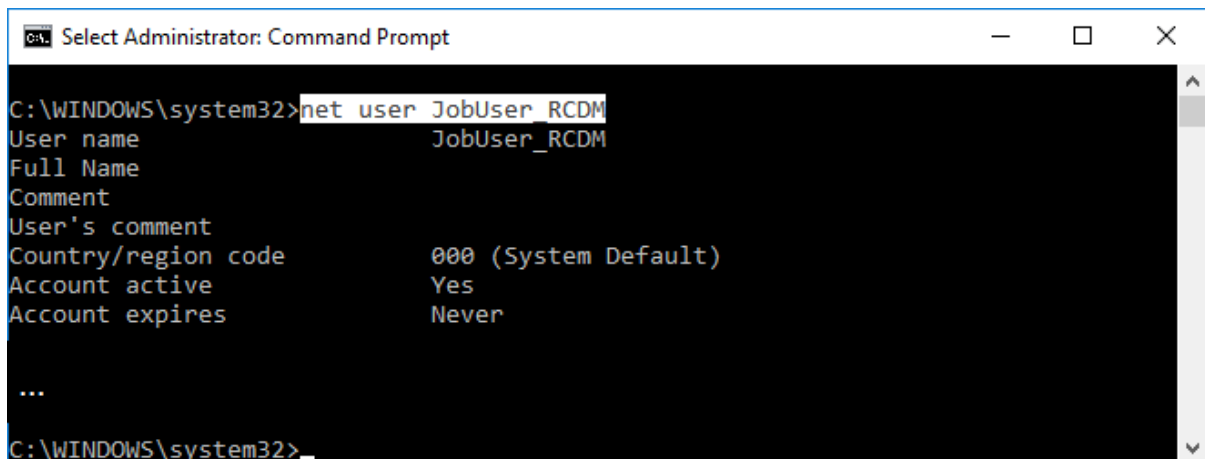
```
net user /add JobUser_RCDM Pa$$w0rd_extraG3heim!!_and_extra$trong
```



```
C:\WINDOWS\system32>net user /add JobUser_RCDM Pa$$w0rd_extraG3heim!!_and_extra$trong
The password entered is longer than 14 characters. Computers
with Windows prior to Windows 2000 will not be able to use
this account. Do you want to continue this operation? (Y/N) [Y]: Y
The command completed successfully.

C:\WINDOWS\system32>
```

Zum Verifizieren ..



```
C:\WINDOWS\system32>net user JobUser_RCDM
User name                JobUser_RCDM
Full Name
Comment
User's comment
Country/region code      000 (System Default)
Account active           Yes
Account expires         Never
...

C:\WINDOWS\system32>
```

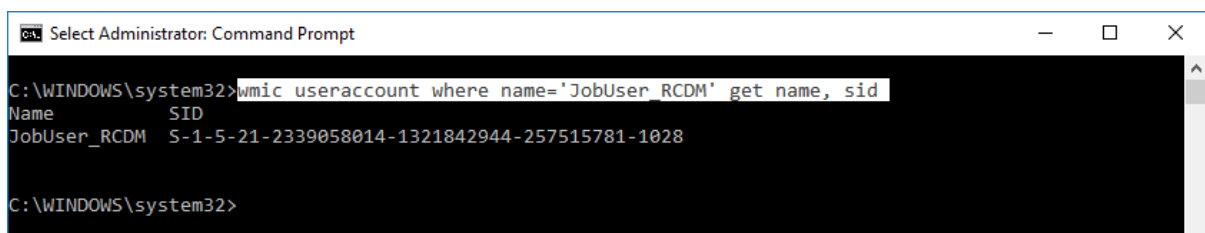
```
net user JobUser_RCDM
```

Der lokale Windows-User ist angelegt.

Für domainbasierte Windows-User

```
net user JobUser_RCDM /domain
```

Alternativ mit WMIC.



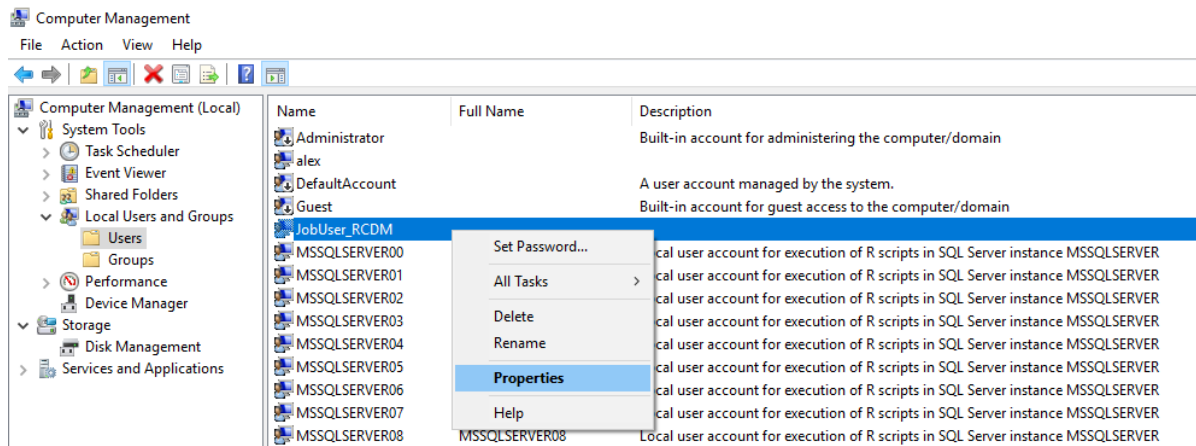
```
C:\WINDOWS\system32>wmic useraccount where name='JobUser_RCDM' get name, sid
Name                SID
JobUser_RCDM      S-1-5-21-2339058014-1321842944-257515781-1028

C:\WINDOWS\system32>
```

```
wmic useraccount where name='JobUser_RCDM' get name, sid
```

Der lokale Windows-User ist angelegt.

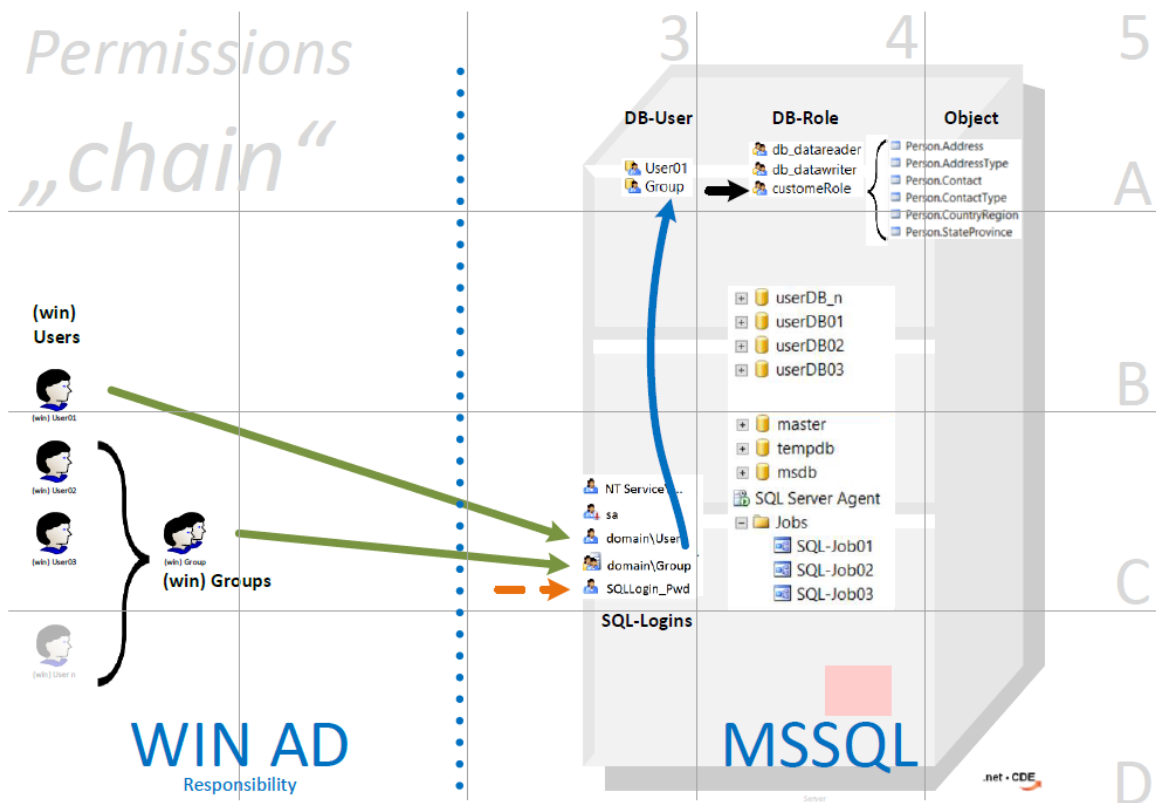
## Eine Administration im Computer Management ...

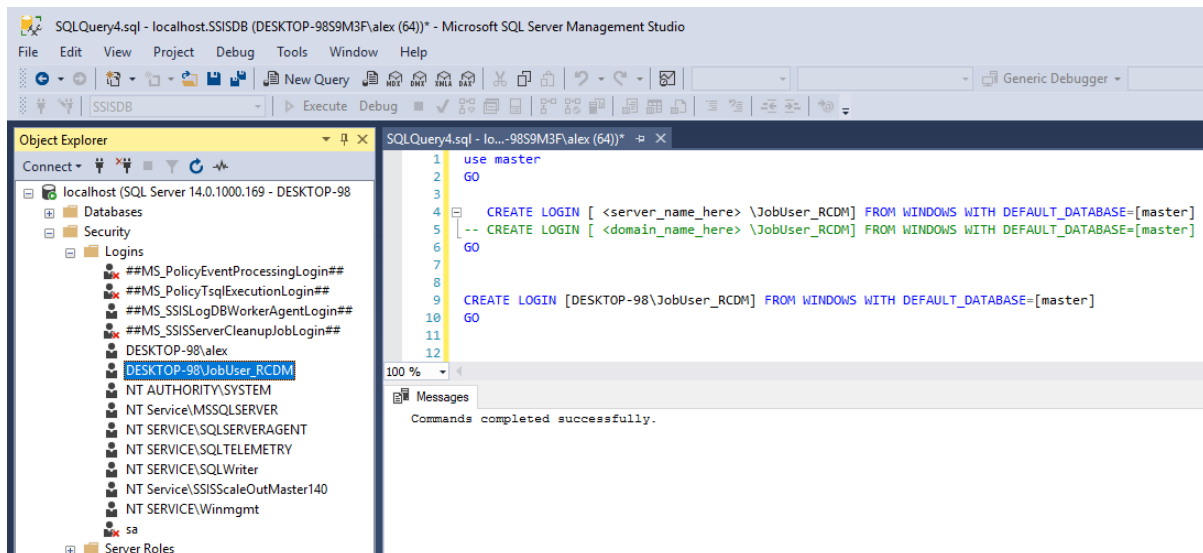


... kann über die zugehörigen Eigenschaften erfolgen.

Jetzt folgt die weitere Verarbeitung wieder im SQL Server Management Studio.

Zuvor eine Übersicht bzgl. „Users & Logins“





```

CREATE LOGIN [ <server_name_here> \JobUser_RCDM] FROM WINDOWS WITH DEFAULT_DATABASE=[master]
-- CREATE LOGIN [ <domain_name_here> \JobUser_RCDM] FROM WINDOWS WITH DEFAULT_DATABASE=[master]

```

Zunächst wird der eben erstellte JobUser berechtigt um in den Server zu gelangen.

```

IF NOT EXISTS( select * from sys.server_principals where name like '%JobUser_RCDM' )
CREATE LOGIN [DESKTOP-98\JobUser_RCDM] FROM WINDOWS WITH DEFAULT_DATABASE=[master]
GO

```

Um Warnungen bzw. Fehlermeldungen vorzubeugen wird den Befehlen ein IF NOT EXISTS als Bedingung vorangestellt.

```

-----
USE [SSISDB]
GO

```

```

IF NOT EXISTS( select * from SSISDB.sys.database_principals where name like
'JobUser_RCDM' )
CREATE USER [JobUser_RCDM] FOR LOGIN [DESKTOP-98\JobUser_RCDM] WITH
DEFAULT_SCHEMA=[catalog]
GO

```

Anschließend wird der Login auch als DB-User in der SSIS Catalog DB angelegt.

```

-- *****

```

Vergleichbar „wie auf C:\“ kann ein Projekt nicht im „Root“ angelegt werden;

Deshalb ist zunächst ein Ordner innerhalb der SSISDB zu erstellen.

```

catalog.create_folder

```

Dann folgt das Erteilen der Ordner-Berechtigungen für den „technischen User“.

Hierbei sind sowohl die folder\_id (Ordner-ID) als auch die principal\_id (Benutzer-ID) aus den jeweiligen System-Views zu ermitteln.

Die zugehörige T-SQL Codierung ist auf der nächsten Seite:



```

-- create Folder
Declare @pFolder_name          nvarchar(128) = N'RCDM_folder'
        , @pFolder_description nvarchar(max) = N'dies ist der RCDM_folder ;'
        , @pFolder_id          bigint
        , @pPrincipal_id       int
        , @pPrincipal_name     sysname = N'JobUser_RCDM'
---

IF NOT EXISTS( select folder_id from SSISDB.catalog.folders where name = @pFolder_name
)
BEGIN
    exec SSISDB.catalog.create_folder @folder_name = @pFolder_name
        , @folder_id = @pFolder_id OUTPUT

    exec SSISDB.catalog.set_folder_description @folder_name = @pFolder_name
        , @folder_description =
@pFolder_description
END -- end IF

-----
-- @object_type
-- Ordner (1)
-- Projekt (2)
-- Umgebung (3)
-- Vorgang (4)

-- @object_type=1 ist fix.
-- @permission_type=1 ist fix.

select @pFolder_id = folder_id
from SSISDB.catalog.folders
where name = @pFolder_name

select @pPrincipal_id = principal_id
from SSISDB.sys.database_principals
where name = @pPrincipal_name

EXEC SSISDB.catalog.grant_permission @object_type = 1 -- Ordner
        , @object_id = @pFolder_id
        , @principal_id = @pPrincipal_id
        , @permission_type = 1 -- READ

EXEC SSISDB.catalog.grant_permission @object_type = 1 -- Ordner
        , @object_id = @pFolder_id
        , @principal_id = @pPrincipal_id
        , @permission_type = 101 -- READ_OBJECTS

EXEC SSISDB.catalog.grant_permission @object_type = 1 -- Ordner
        , @object_id = @pFolder_id
        , @principal_id = @pPrincipal_id
        , @permission_type = 103 -- EXECUTE_OBJECTS

-- https://docs.microsoft.com/de-de/sql/integration-services/system-stored-
-- procedures/catalog-grant-permission-ssisdb-database?view=sql-server-2017

-----
----- *****

```

Die oben erstellte Konfiguration diene dazu innerhalb der SSIS Catalog DB dem „technischen User“ die Berechtigungen für die jeweiligen Projektordner zu erteilen.

Damit auch eine automatische ( d.h. vom SQL Server Agent ausgeführt) Paketausführung möglich ist sind die beiden folgenden Schritte notwendig.

Zunächst werden die Anmeldeinformationen (d.h. Windows-Name und zugehöriges Passwort) in der master-Datenbank eingetragen.

```
USE [master]
GO

IF NOT EXISTS( select * from sys.credentials where name = N'cred_JobUser_RCDM' )
CREATE CREDENTIAL cred_JobUser_RCDM
WITH IDENTITY = N'DESKTOP-98\JobUser_RCDM'
, SECRET = N'<put_Pa$$w0rd_here>'
GO
```

Anschließend erfolgt das Anlegen eines Proxies mit den eben erstellten Anmeldeinformationen in der msdb-Datenbank (= „System-Datenbank des SQL Server Agents). Dabei dient die Zuordnung des Proxies zu einem Subsystem dazu, die Verwendung auf z.B. die Integration Services zu limitieren.

```
USE [msdb]
GO

IF NOT EXISTS( select * from msdb.dbo.sysproxies where name = N'prox_JobUser_RCDM' )
EXEC msdb.dbo.sp_add_proxy @proxy_name = N'prox_JobUser_RCDM'
, @credential_name = N'cred_JobUser_RCDM'
, @enabled = 1
GO

IF EXISTS( select * from msdb.dbo.sysproxies where name = N'prox_JobUser_RCDM' )
BEGIN

IF NOT EXISTS ( select *
from msdb.dbo.sysproxies prx
join msdb.dbo.sysproxysubsystem pss
on prx.proxy_id = pss.proxy_id
where prx.name = N'prox_JobUser_RCDM'
and pss.subsystem_id = 11 ) -- 11 = SSIS
EXEC msdb.dbo.sp_grant_proxy_to_subsystem @proxy_name=N'prox_JobUser_RCDM'
, @subsystem_id=11 -- 11 = SSIS

END -- end IF
GO
```

Die gesamte Auflistung der SQL-Agent Subsystem ist hier beschrieben ..

```
-- -- sp_grant_proxy_to_subsystem >> subsystem
-- https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-grant-proxy-to-subsystem-transact-sql?view=sql-server-2017
```

```
-----
----- *****
```

Das gesamte T-SQL script ist in der Datei ..

```
>> deployment_script_part_01.sql
```

>> jetzt folgt die Paket-Entwicklung

mit dieser Empfehlung

[https://testfabrikstorage001.blob.core.windows.net/transport/video\\_database-lifecycle-mgmt\\_mid.mp4](https://testfabrikstorage001.blob.core.windows.net/transport/video_database-lifecycle-mgmt_mid.mp4)

[https://testfabrikstorage001.blob.core.windows.net/transport/video\\_modern\\_DB\\_dev\\_mit\\_SSDT.mp4](https://testfabrikstorage001.blob.core.windows.net/transport/video_modern_DB_dev_mit_SSDT.mp4)

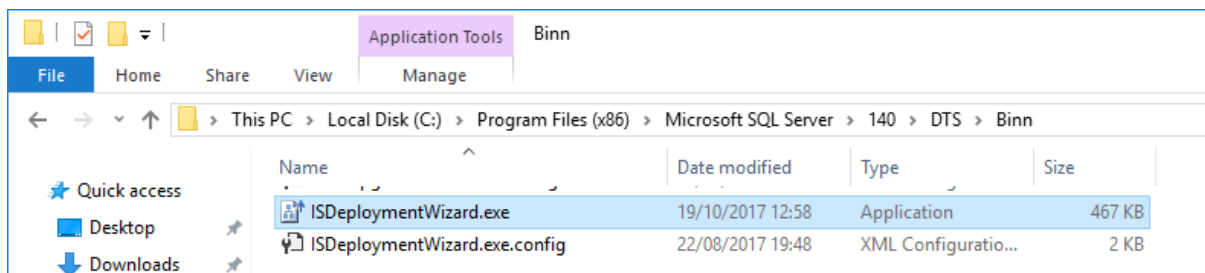
2 Video- bzw. Konferenz-Sessions zum Thema Application Lifecycle von Kostja Klein

mit anschließendem Deployment

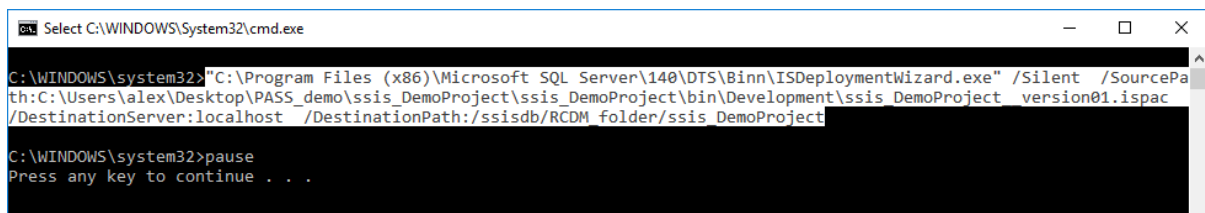
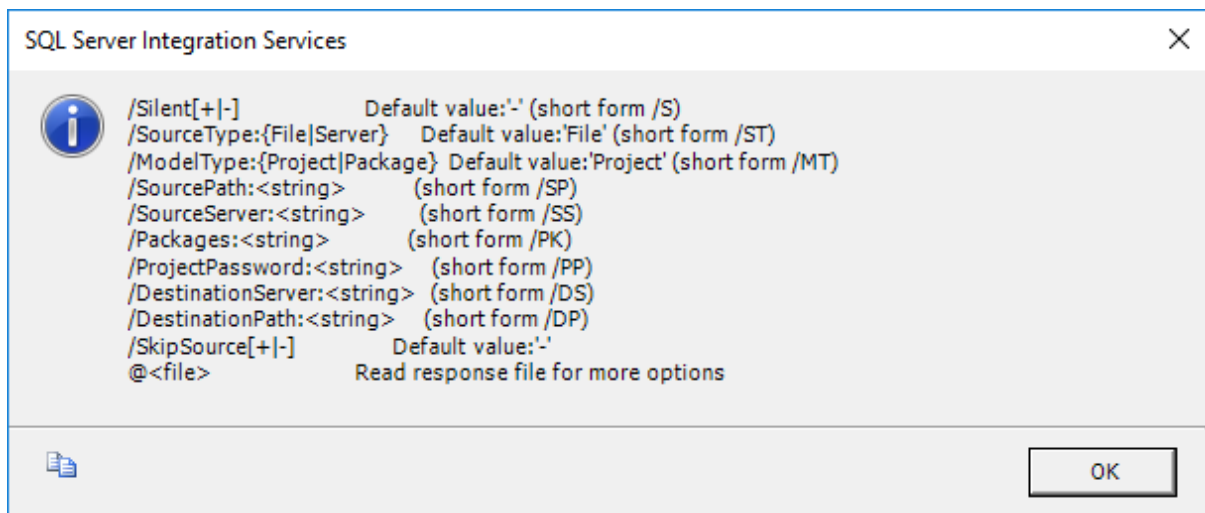
<https://docs.microsoft.com/de-de/sql/integration-services/packages/deploy-integration-services-ssis-projects-and-packages?view=sql-server-2017>

Dafür stehen 2 Optionen zur Wahl:

### 1. ISDeploymentWizard.exe als Commandline-Tool



mit den folgenden Parametern



```
"C:\Program Files (x86)\Microsoft SQL Server\140\DTS\Binn\ISDeploymentWizard.exe"
/Silent
/SourcePath:C:\ <sourceFolder > \ssis_DemoProject__version01.ispac
/DestinationServer:localhost
/DestinationPath:/SSISDB/RCDM_folder/ssis_DemoProject
```

Zu Bedenke ist hier der „Programm-Wechsel“ von T-SQL im SSMS hin zu einem Commandline-Tool.

## 2. mit T-SQL

```
DECLARE @ProjectBinary as varbinary(max)
DECLARE @operation_id as bigint
```

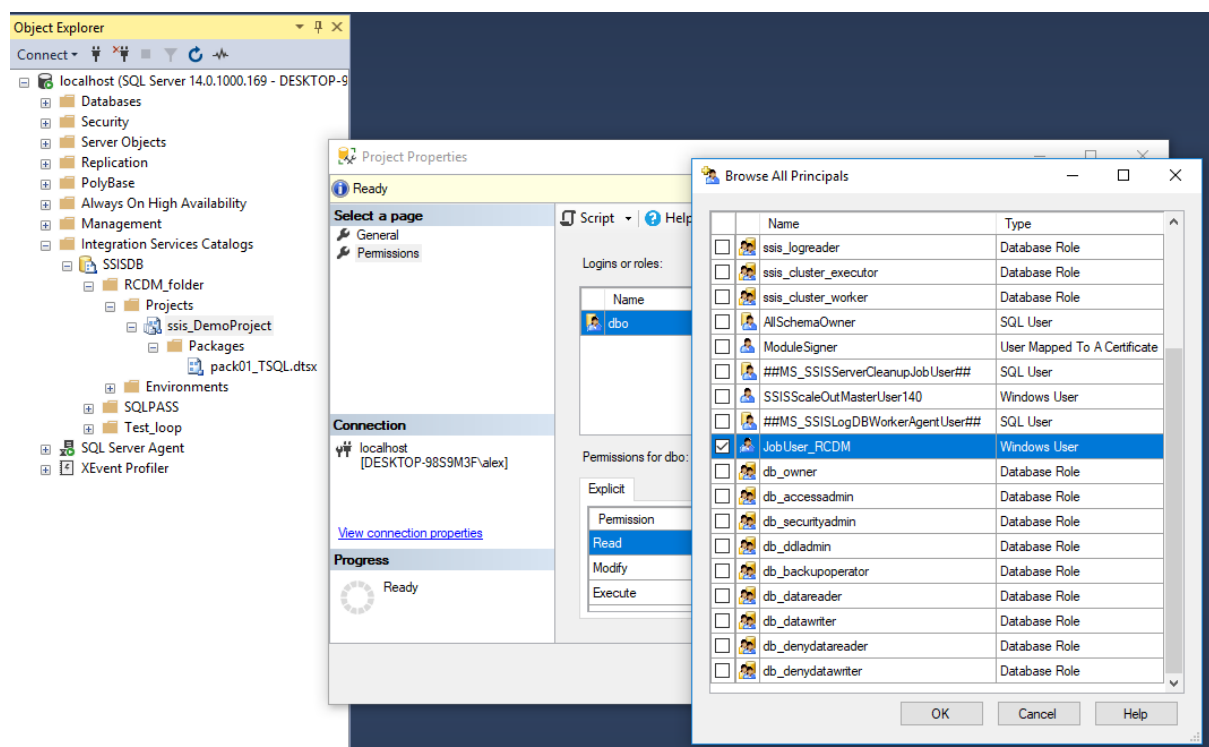
```
Set @ProjectBinary =
(
  SELECT *
  FROM OPENROWSET( BULK 'C:\_tempDeploy\ssis_DemoProject__version01.ispac'
    , SINGLE_BLOB) as BinaryData)
```

```
Exec SSISDB.catalog.deploy_project @folder_name = 'RCDM_folder'
    , @project_name = 'ssis_DemoProject'
    , @Project_Stream = @ProjectBinary
    , @operation_id = @operation_id out
```

Hier dagegen gilt es zu Bedenken, dass das Deployment-Team „Bulk-Insert“ Permissions auf Serverebene benötigt.

<https://docs.microsoft.com/en-us/sql/integration-services/packages/deploy-integration-services-ssis-projects-and-packages?view=sql-server-2017>

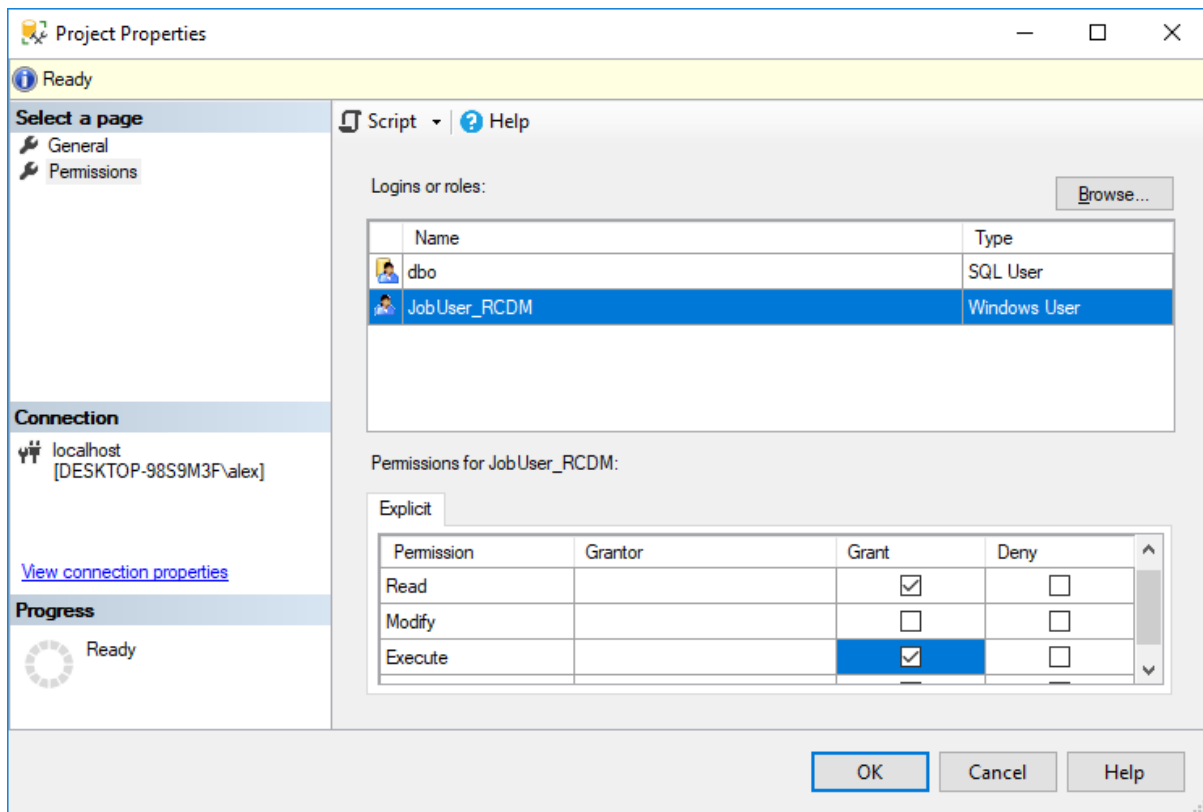
Zusätzlich ist wieder die Ausführungsberichtigung (auf Project-Ebene) zu erteilen.



Ergänzend zum `SSISDB.catalog.grant_permission` (vgl. Seite 9) erfolgt hier die Konfiguration im Kontextmenü. Die Projekt-Eigenschaften enthalten einen Menüpunkt für die Berechtigungen. Standardmäßig sind dort sämtliche Berechtigungen für den Principal `dbo` gesetzt.

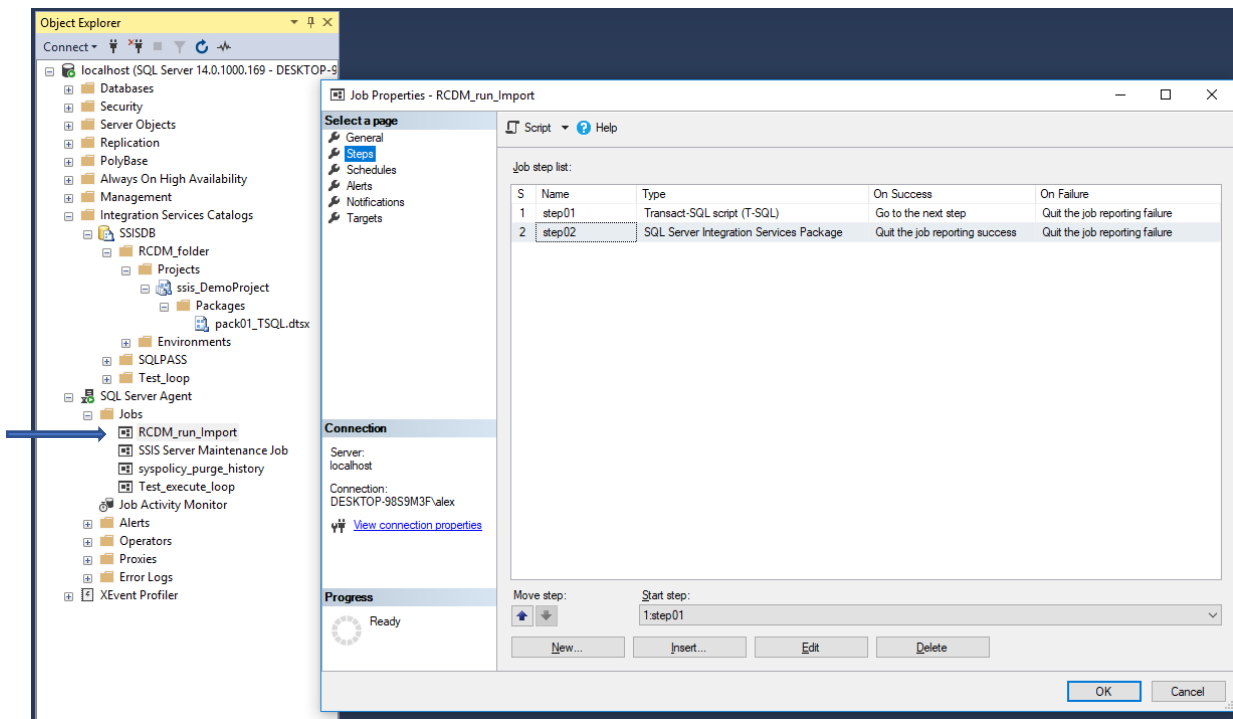
Ein weiterer Principal (`JobUser_RCDM`) ist mittels [Browse...] auszuwählen. Dann können diesem Principal die Berechtigungen `Read` und `Execute` erteilt werden.

Sollte der zu berechtigende Principal ( im Beispiel der JobUser\_RCDM ) nicht in der Auflistung erscheinen, ist der jeweilige Principal nicht in der SSIDB als Datenbank-User angelegt.  
( vgl. Seite 8 `CREATE USER [JobUser_RCDM] ...` )



## Create and config Environment

Jetzt folgt die Konfiguration im SQL Server Agent.



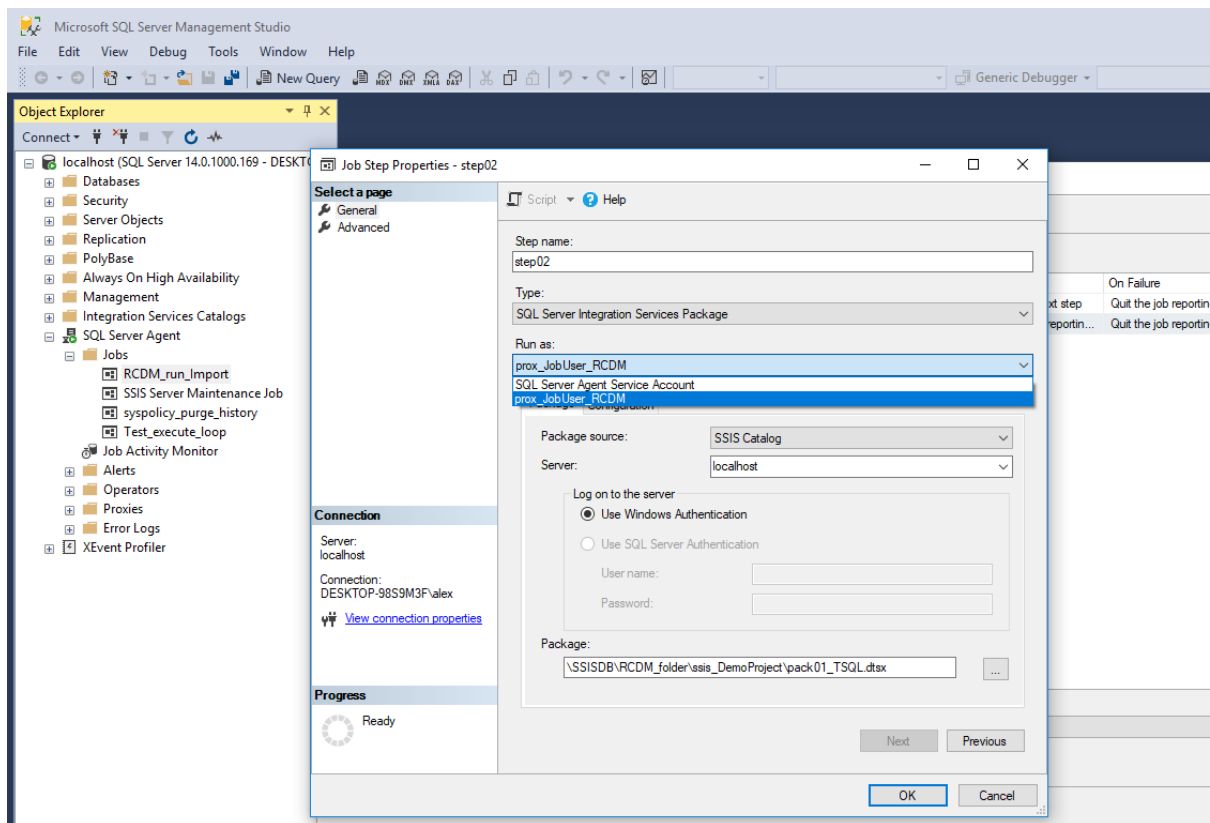
Im jeweiligen Kontext-Menü eines Jobs lassen sich die zugehörigen Eigenschaften auswählen bzw. anpassen. Dabei kann ein Job als Containment für die auszuführenden Job-Steps gesehen werden. Im Beispiel sind step01 ( vom Typ Transact-SQL ) und step02 ( vom Typ SQL Server Integration Services Package ) zu sehen.

In der Job step list sind die Job-Steps in ihrer Ausführungsreihenfolge aufgelistet. Weiterhin ist in den Spalten [On Success] und [On Error] auch konfiguriert, was nach erfolgreicher bzw. fehlgeschlagener Ausführung eines Job-Steps folgt. Die Standards sind hier beispielhaft dargestellt und konfiguriert; weitere Möglichkeiten sind, die gesamte Job-Ausführung fortzusetzen auch wenn ein Step fehlschlägt oder auch bedingte Sprünge zu nachfolgend konfigurierten Job-Steps.

Die weiteren Job-Eigenschaften der Zeitpläne, Warnungen, Benachrichtigungen und Ziele sind hier nicht Gegenstand der Beschreibung; Informationen dazu liefert Microsoft hier <https://docs.microsoft.com/de-de/sql/ssms/agent/job-step-properties-new-job-step-general-page?view=sql-server-2017>



Jetzt folgt die Job-Step Konfiguration am Beispiel eines SSIS-Tasks.



Zunächst ist der Ausführungs Type auf SQL Server Integration Services Package zu stellen. Damit werden dann auch die SSIS-typischen Konfigurationsfelder sichtbar.

Dann folgt die Konfiguration „Run as“, d.h. der Ausführungskontext des jeweiligen Steps. Zur Auswahl stehen das Konto des SQL Server Agents und der oben zusätzlich konfigurierte Proxy. Damit werden die Paketausführungen ausschließlich mit den Berechtigungen des Windows Users „JobUser\_RCDM“ ausgeführt. Berechtigungsänderungen für die Job-Step Konfiguration können demnach dediziert diesem User zukommen und beeinflussen nicht sämtliche weitere Jobs.

Fehlende Berechtigungen sind in den zugehörigen Logs ebenfalls dem User „JobUser\_RCDM“ zuzuordnen. Dies ist in dem Beispiel auf Seite 21 und 22 dargestellt.

Das Konto des SQL Server Agent Service Accounts erfährt dagegen nach der Erstinstallation keine weitere Konfigurationsänderung.

Dann folgend die SSIS-spezifischen Konfigurationen beginnend mit der Package Source.

Die bevorzugte Quelle ist der SSIS Catalog.

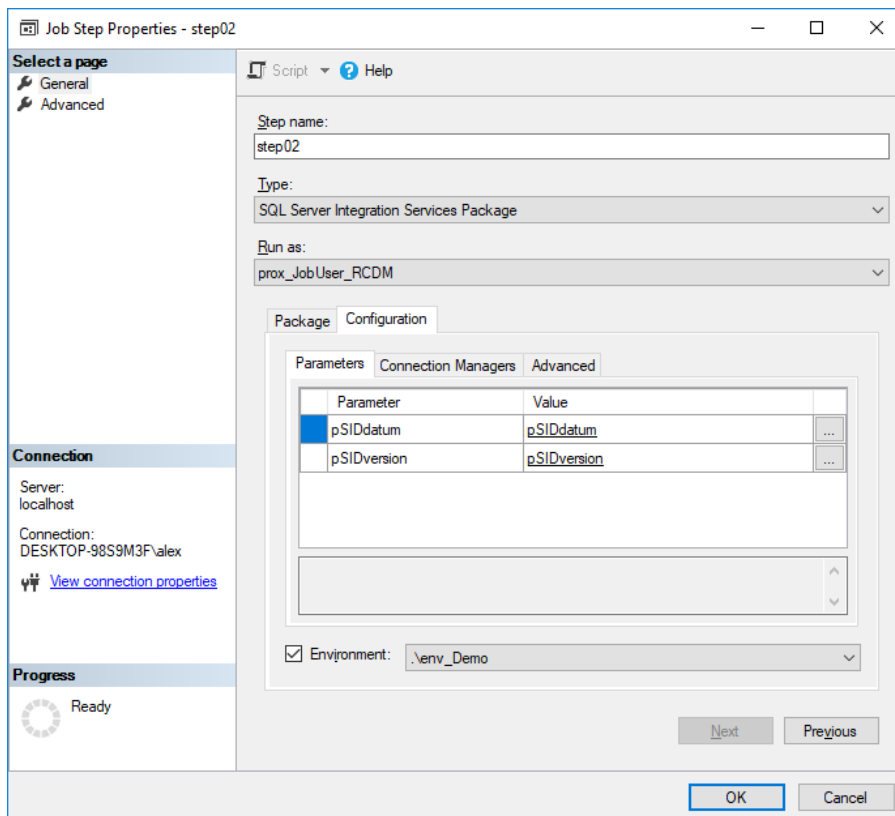
Die weiteren verfügbaren Quellen sind hier

<https://docs.microsoft.com/en-us/sql/integration-services/packages/sql-server-agent-jobs-for-packages?view=sql-server-2017>

>> 8. Package Source beschrieben.

Die drei folgenden Einstellungen [Server], [Log on to the Server] und [Package] setzen die oben erteilten Berechtigungen ( **CREATE LOGIN** und `SSISDB.catalog.grant_permission` für Ordner und Project ) voraus.

Im nächsten Register



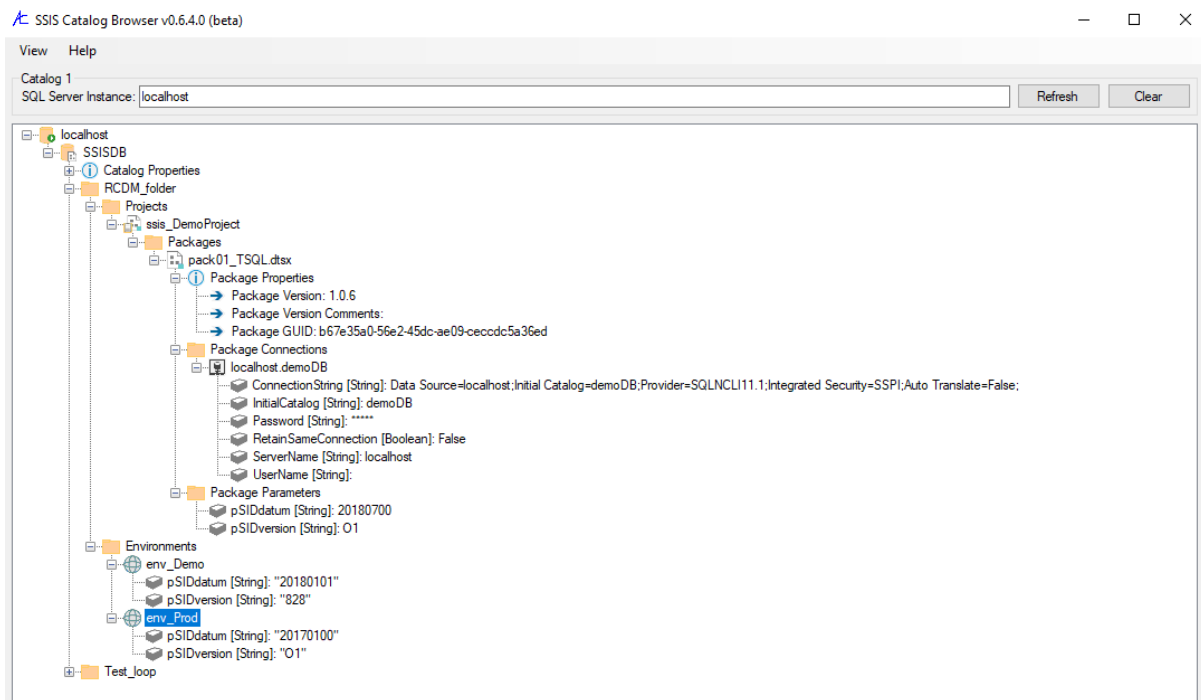
Folgen die Konfigurationen für die Parameters und Connection Managers. Weiterhin sind in Advanced der Logging level und die Ausführung in der 32-bit runtime wählbar.

Im vorliegenden Beispiel erfolgte die Konfiguration der Parameter mittels Environment.

Die dabei konfigurierten Werte sind innerhalb der Environments den Variablen zugeordnet.

Um also die zur Laufzeit verwendeten Werte für pSIDdatum bzw. sSIDversion zu erfahren ist ein Blick in die Eigenschaften des Environments ( im Beispiel „env\_Demo“ ) notwendig.

Eine schnellere Übersicht bzgl. der Parameter und Config-Strings bietet der SSIS Catalog Browser.

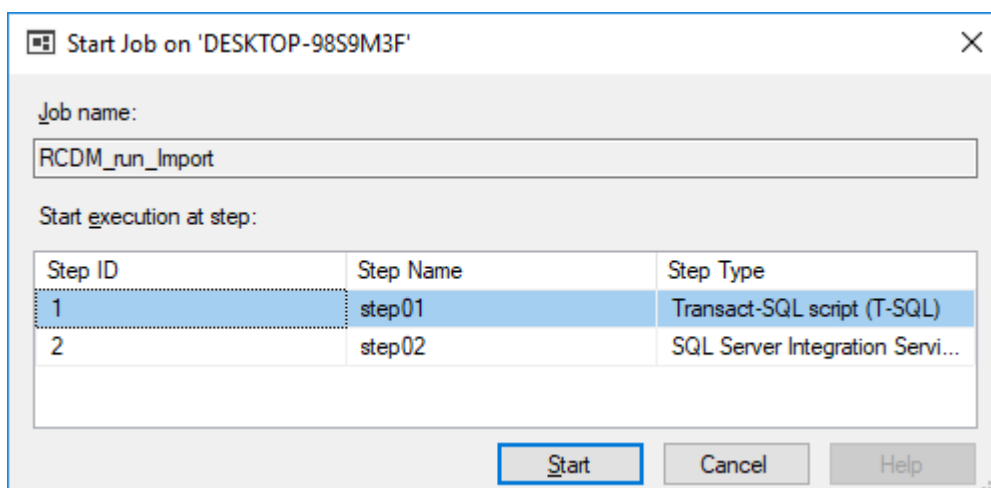


Dann folgt die Ausführung des Jobs.

Dies kann zum einen zeitgesteuert erfolgen. Dies ist in der Job-History am Text „The Job was invoked by Schedule“ erkennbar.

Zum zweiten kann der Job auch „von Hand“ gestartet werden.

Dabei erlaubt der folgende Dialog auch den ersten Step, mit dem gestartet werden soll auszuwählen.



Zu beachten ist, dass der Job bei nur einem Job-Step sofort und ohne Anzeige dieses Dialogs startet.

Die dritte Möglichkeit ist das programmgesteuerte Anstarten mittels T-SQL Statement.

```
exec msdb.dbo.sp_start_job @job_name = N'RCDM_run_Import'
```

Nach der Ausführung folgt die Auswertung in der SQL Server Agent Job-History und in den SSIS Catalog Reports.

```

1 use msdb
2 go
3
4 select top(50) j.name as 'JobName'
5     , h.*
6 from   msdb.dbo.sysjobs j
7 join   msdb.dbo.sysjobhistory h
8 on     j.job_id = h.job_id
9 order by instance_id desc
10

```

JobName	instance_id	job_id	step_id	step_name	sql_message_id	sql_severity	message
RCDM_run_Import	1069	9DF8DBBE-7AC3-49E0-A6F3-A858245E0A05	0	(Job outcome)	0	0	The job succeeded. The Job was invoked by User DESK...
RCDM_run_Import	1068	9DF8DBBE-7AC3-49E0-A6F3-A858245E0A05	1	step01	0	0	Executed as user: NT SERVICE\SQLSERVERAGENT, The...
RCDM_run_Import	1067	9DF8DBBE-7AC3-49E0-A6F3-A858245E0A05	0	(Job outcome)	0	0	The job succeeded. The Job was invoked by User DESK...
RCDM_run_Import	1066	9DF8DBBE-7AC3-49E0-A6F3-A858245E0A05	1	step01	0	0	Executed as user: NT SERVICE\SQLSERVERAGENT, The...
RCDM_run_Import	1065	9DF8DBBE-7AC3-49E0-A6F3-A858245E0A05	0	(Job outcome)	0	0	The job succeeded. The Job was invoked by User DESK...
RCDM_run_Import	1064	9DF8DBBE-7AC3-49E0-A6F3-A858245E0A05	1	step01	0	0	Executed as user: NT SERVICE\SQLSERVERAGENT, The...

Die folgende Auswertung joint den Jobnamen mit sämtlichen Jobhistory-Details.

Insbesondere sind die Spalten run\_time und run\_duration zu beachten. Der jeweilige Wert ist die „unformatierte“ Uhrzeit bzw. Ausführungsdauer. Die formatierenden Doppelpunkte sind zugunsten des Datentyps INT herstellerseitig entfernt.

```

select top(50) j.name as 'JobName'
    , h.*
from   msdb.dbo.sysjobs j
join   msdb.dbo.sysjobhistory h
on     j.job_id = h.job_id
order  by instance_id desc

```

Ein Eintrag in die sysjobhistory erfolgt erst, nachdem der erste Jobstep beendet ist. Demnach sind „langlaufende erster Jobsteps“ zunächst hier nicht sichtbar.

```

15
16 SELECT
17     ja.job_id,
18     j.name AS job_name,
19     ja.start_execution_date,
20     ISNULL(last_executed_step_id,0)+1 AS current_executed_step_id,
21     js.step_name
22 FROM msdb.dbo.sysjobactivity ja
23 LEFT JOIN msdb.dbo.sysjobhistory jh
24     ON ja.job_history_id = jh.instance_id
25 JOIN msdb.dbo.sysjobs j
26     ON ja.job_id = j.job_id
27 JOIN msdb.dbo.sysjobsteps js
28     ON ja.job_id = js.job_id
29     AND ISNULL(ja.last_executed_step_id,0)+1 = js.step_id
30 WHERE ja.session_id = (SELECT TOP 1 session_id FROM msdb.dbo.syssessions ORDER BY agent_start_date DESC)
31 AND start_execution_date is not null
32 AND stop_execution_date is null;

```

job_id	job_name	start_execution_date	current_executed_step_id	step_name
9DF8DBBE-7AC3-49E0-A6F3-A858245E0A05	RCDM_run_Import	2018-07-10 14:33:27.000	1	step01

Diese Abfrage zeigt die vermeintlich fehlende Information.

SELECT

```
ja.job_id,  
j.name AS job_name,  
ja.start_execution_date,  
ISNULL(last_executed_step_id,0)+1 AS current_executed_step_id,  
Js.step_name
```

FROM msdb.dbo.sysjobactivity ja

LEFT JOIN msdb.dbo.sysjobhistory jh

```
ON ja.job_history_id = jh.instance_id
```

JOIN msdb.dbo.sysjobs j

```
ON ja.job_id = j.job_id
```

JOIN msdb.dbo.sysjobsteps js

```
ON ja.job_id = js.job_id
```

```
AND ISNULL(ja.last_executed_step_id,0)+1 = js.step_id
```

WHERE ja.session\_id = (SELECT TOP 1 session\_id FROM msdb.dbo.syssessions ORDER BY agent\_start\_date DESC)

AND start\_execution\_date is not null

AND stop\_execution\_date is null;

Die zugehörige Ansicht im Log File Viewer

The screenshot shows the Log File Viewer interface for a SQL Server job named 'RCDM\_run\_Import'. The main table displays the following data:

Date	Step ID	Server	Job Name	Step Name	Notifications	Message
10/07/2018 14:39:17	2	DESKTOP-98S9M3F	RCDM_run_Import	step02		The job failed. The Job was invoked by User DESKTOP-98S9M3F\alex. The last step to run was step 2 (step02). The job was requested to start at step 1 (step01).
10/07/2018 14:41:29	1	DESKTOP-98S9M3F	RCDM_run_Import	step01		Executed as user: DESKTOP-98S9M3F\JobUser_RCDM. Microsoft (R) SQL Server Execute Package Utility Version 14.0.1000.16
10/07/2018 14:33:27		DESKTOP-98S9M3F	RCDM_run_Import			Executed as user: NT SERVICE\SQLSERVERAGENT. The step succeeded.
10/07/2018 14:30:41		DESKTOP-98S9M3F	RCDM_run_Import			The job succeeded. The Job was invoked by User DESKTOP-98S9M3F\alex. The last step to run was step 1 (step01).

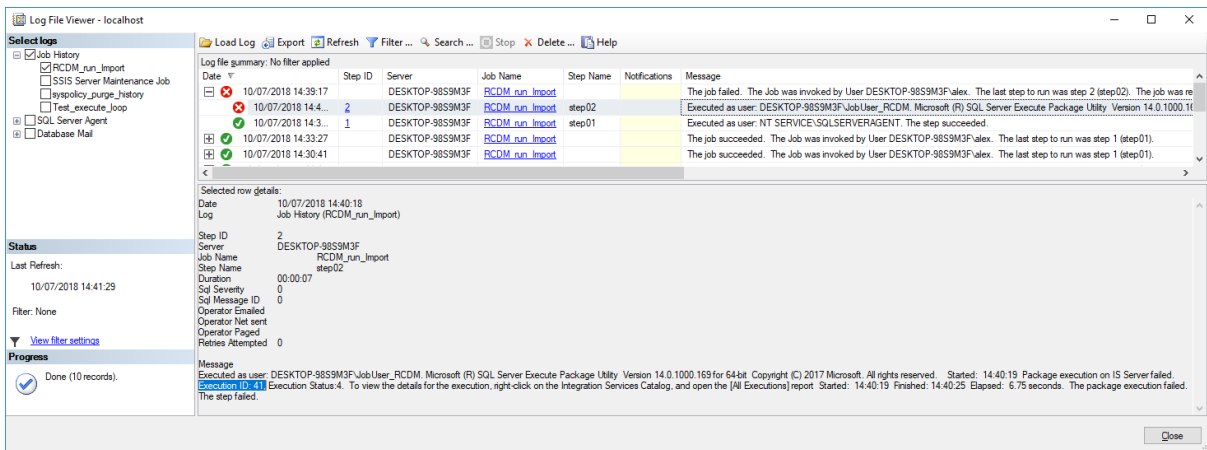
Below the table, the 'Selected row details' section shows the following information for the failed step:

- Date: 10/07/2018 14:39:17
- Log: Job History (RCDM\_run\_Import)
- Step ID: 2
- Server: DESKTOP-98S9M3F
- Job Name: RCDM\_run\_Import
- Step Name: step02
- Duration: 00:01:08
- Sql Severity: 0
- Sql Message ID: 0
- Operator Emailed: 0
- Operator Net sent: 0
- Operator Paged: 0
- Retries Attempted: 0

The 'Progress' section shows 'Done (10 records)' with a checkmark icon.

Die Job-Message:

*The job failed. The Job was invoked by User DESKTOP-98S9M3F\alex. The last step to run was step 2 (step02). The job was requested to start at step 1 (step01).*



Die Message des Job-Steps:

*Executed as user: DESKTOP-98S9M3F\JobUser\_RCDM. Microsoft (R) SQL Server Execute Package Utility Version 14.0.1000.169 for 64-bit Copyright (C) 2017 Microsoft. All rights reserved. Started: 14:40:19 Package execution on IS Server failed. Execution ID: 41, Execution Status:4. To view the details for the execution, right-click on the Integration Services Catalog, and open the [All Executions] report Started: 14:40:19 Finished: 14:40:25 Elapsed: 6.75 seconds. The package execution failed. The step failed.*

Mit dieser . **Execution ID: 41** kann die weitere Fehler-Analyse in den Reports der SSISDB fortgesetzt werden.

Die Übersicht aller Ausführungen:

### All Executions

on DESKTOP-98S9M3F at 7/10/2018 2:46:34 PM

This report provides information about the Integration Services package executions that have been performed on the connected SQL Server instance.

**Filter:** Start time range: 7/4/2018 - 7/10/2018; Status: All; (4 more)

Execution Information

2
0
1
0

Failed
Running
Succeeded
Others

ID	Status	Report	Folder Name	Project Name	Package Name
41	Failed	<a href="#">Overview</a> <a href="#">All Messages</a> <a href="#">Execution Performance</a>	RCDM_folder	ssis_DemoProject	pack01_TSQL.dtsx
40	Succeeded	<a href="#">Overview</a> <a href="#">All Messages</a> <a href="#">Execution Performance</a>	RCDM_folder	ssis_DemoProject	pack01_TSQL.dtsx
28	Failed	<a href="#">Overview</a> <a href="#">All Messages</a> <a href="#">Execution Performance</a>	RCDM_folder	ssis_DemoProject	pack01_TSQL.dtsx

1 / 1

## Messages

on DESKTOP-98S9M3F at 7/10/2018 2:47:22 PM

This report displays messages for the executions or validations.

[View Overview](#)

### Execution Information

Operation ID	41 (Execution)
Name	RCDM_folder\ssis_DemoProject\pack01_TSQL.dtsx
Environment	-
Status	Failed

Duration (sec)	2.766
Start Time	7/10/2018 2:40:22 PM
End Time	7/10/2018 2:40:24 PM
Caller	DESKTOP-98S9M3F\JobUser_RCDM

### Error Messages

Message Type	Action	Message Time	Message	Message Source Name
OnError	<a href="#">View Context</a>	7/10/2018 2:40:24 PM	SQL_CAST_and_Calc:Error: Failed to acquire connection "localhost.demoDB". Connection may not be configured correctly or you may not have the right permissions on this connection.	SQL_CAST_and_Calc
OnError	<a href="#">View Context</a>	7/10/2018 2:40:24 PM	pack01_TSQL:Error: SSIS Error Code DTS_E_OLEDBBERROR. An OLE DB error has occurred. Error code: 0x80004005. An OLE DB record is available. Source: "Microsoft SQL Server Native Client 11.0" Hresult: 0x80004005 Description: "Login failed for user 'DESKTOP-98S9M3F\JobUser_RCDM'." An OLE DB record is available. Source: "Microsoft SQL Server Native Client 11.0" Hresult: 0x80004005 Description: "Cannot open database 'demoDB' requested by the login. The login failed.".	pack01_TSQL

Nach dem Fixen des Berechtigungsproblems ...

The screenshot shows the 'Log File Viewer - localhost' window. The left pane shows a tree view with 'Job History' expanded, containing 'RCDM\_run\_Import', 'SSIS Server Maintenance Job', 'syspolicy\_purge\_history', and 'Test\_execute\_Joop'. The main pane displays a table of log entries:

Date	Step ID	Server	Job Name	Step Name	Notifications	Message
10/07/2018 14:53:17		DESKTOP-98S9M3F	RCDM_run_Import			The job failed. The Job was invoked by User DESK...
10/07/2018 14:53:18	2	DESKTOP-98S9M3F	RCDM_run_Import	step02		Executed as user: DESKTOP-98S9M3F\JobUser_R...
10/07/2018 14:53:18	1	DESKTOP-98S9M3F	RCDM_run_Import	step01		Executed as user: NT SERVICE\SQLSERVERAGE...

Below the table, 'Selected row details' are shown for the selected row (Step ID 2):

- Date: 10/07/2018 14:53:18
- Log: Job History (RCDM\_run\_Import)
- Step ID: 2
- Server: DESKTOP-98S9M3F
- Job Name: RCDM\_run\_Import
- Step Name: step02
- Duration: 00:00:09
- Sql Severity: 0
- Sql Message ID: 0
- Operator Emailed:
- Operator Net sent:
- Operator Paged:
- Retries Attempted: 0

The 'Status' section shows the message: 'Executed as user: DESKTOP-98S9M3F\JobUser\_RCDM, Microsoft (R) SQL Server Execute Package Utility Version 14.0.1000.169 for 64-bit. Copyright (C) 2017 Microsoft. All rights reserved. Started: 14:53:18 Package execution on IS Server failed. Execution ID: 44. Execution Status: 4. To view the details for the execution, right-click on the Integration Services Catalog, and open the [All Executions] report. Started: 14:53:18 Finished: 14:53:27 Elapsed: 8.703 seconds. The package execution failed. The step failed.'

## Die Details im SSIS Catalog zeigen..

Execution Information

Operation ID	44 (Execution)	Duration (sec)	2.406
Name	RCDM_folder\ssis_DemoProject\pack01_TSQL.dtsx	Start Time	7/10/2018 2:53:22 PM
Environment	-	End Time	7/10/2018 2:53:25 PM
Status	Failed	Caller	DESKTOP-98S9M3F\JobUser_RCDM

☒ Error Messages

Message Type	Action	Message Time	Message	Message Source Name
OnError	<a href="#">View Context</a>	7/10/2018 2:53:24 PM	SQL_CAST_and_Calc:Error: Executing the query "Declare @pSIDdatum varchar(50) = ? ..." failed with the following error: "Conversion failed when converting the varchar value '01' to data type int.". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.	SQL_CAST_and_Calc
OnError	<a href="#">View Context</a>	7/10/2018 2:53:24 PM	SQL_CAST_date:Error: Executing the query "Declare @pSIDdatum varchar(50) = ?  SELECT CAS..." failed with the following error: "Conversion failed when converting date and/or time from character string.". Possible failure reasons: Problems with the query, "ResultSet" property not set correctly, parameters not set correctly, or connection not established correctly.	SQL_CAST_date

## Wie kann ein Administrator die zur Laufzeit konfigurierten Parameter auslesen?

SSIS Catalog Browser v0.6.4.0 (beta)

View Help

Catalog 1  
SQL Server Instance: localhost

The screenshot shows the SSIS Catalog Browser interface. The tree view is expanded to show the 'Package Parameters' for the package 'pack01\_TSQL.dtsx'. The parameters listed are:

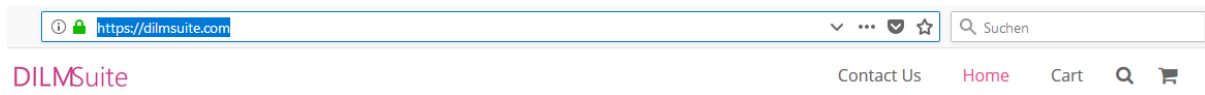
- ConnectionString [String]: Data Source=localhost;Initial Catalog=demoDB;Provider=SQLNCLI11.1;Integrated Security=SSPI;Auto Translate=False;
- InitialCatalog [String]: demoDB
- Password [String]: \*\*\*\*\*
- RetainSameConnection [Boolean]: False
- ServerName [String]: localhost
- UserName [String]:
- pSIDdatum [String]: 20180700
- pSIDversion [String]: 01

Below the package parameters, the 'Environments' folder is also expanded, showing the values for the 'env\_Prod' environment:

- pSIDdatum [String]: "20170100"
- pSIDversion [String]: "01"



Das Tool gibt es zum Download hier <https://dilmsuite.com/>



## “Can I use *Power BI* to view SSIS Catalog execution logs?”

### Basic SSIS Catalog Dashboard in Power BI

Basic SSIS Catalog Dashboard in Power BI is a free Power BI template. Basic SSIS Catalog Dashboard in Power BI is a simple Catalog dashboard with slicers for Date components and SSIS Catalog projects and packages.

## “Can I view *all* of the SSIS Catalog at a glance?”

### SSIS Catalog Browser

SSIS Catalog Browser is a free utility. SSIS Catalog Browser is a single-treeview Catalog viewer, similar to [SSIS Catalog Compare](#). SSIS Catalog Browser presents a rich, complete view of the SSIS projects, packages, and configurations metadata deployed to enterprise SSIS Catalog instances.

... und für die Ausführungsperformance gibt es ein PowerBI dashboard

