



Wie eine SQL Server-Datenbank deployen?



Thorsten Kansy (tkansy@dotnetconsulting.eu)

Meine Person- Thorsten Kansy

Freier Consultant, Software Architekt,
Entwickler, Trainer & Fachautor



Mein Service

- Individuelle Inhouse Trainings
- (Online on-demand) Projektbegleitung
- Beratung
 - Problemanalyse und Lösungen
 - Technologieentscheidungen



Agenda

- Verwendete Software
- Übersicht über die Optionen
- Vor- & Nachteile dieser Optionen

A photograph of two farmers working in a flooded rice paddy. The farmer on the left is wearing a pink and white patterned shirt and a traditional conical hat. The farmer on the right is wearing a yellow shirt and a similar conical hat. They are both bent over, working in the shallow water. The background shows a vast expanse of green rice plants under a clear sky.

Verwendete Software



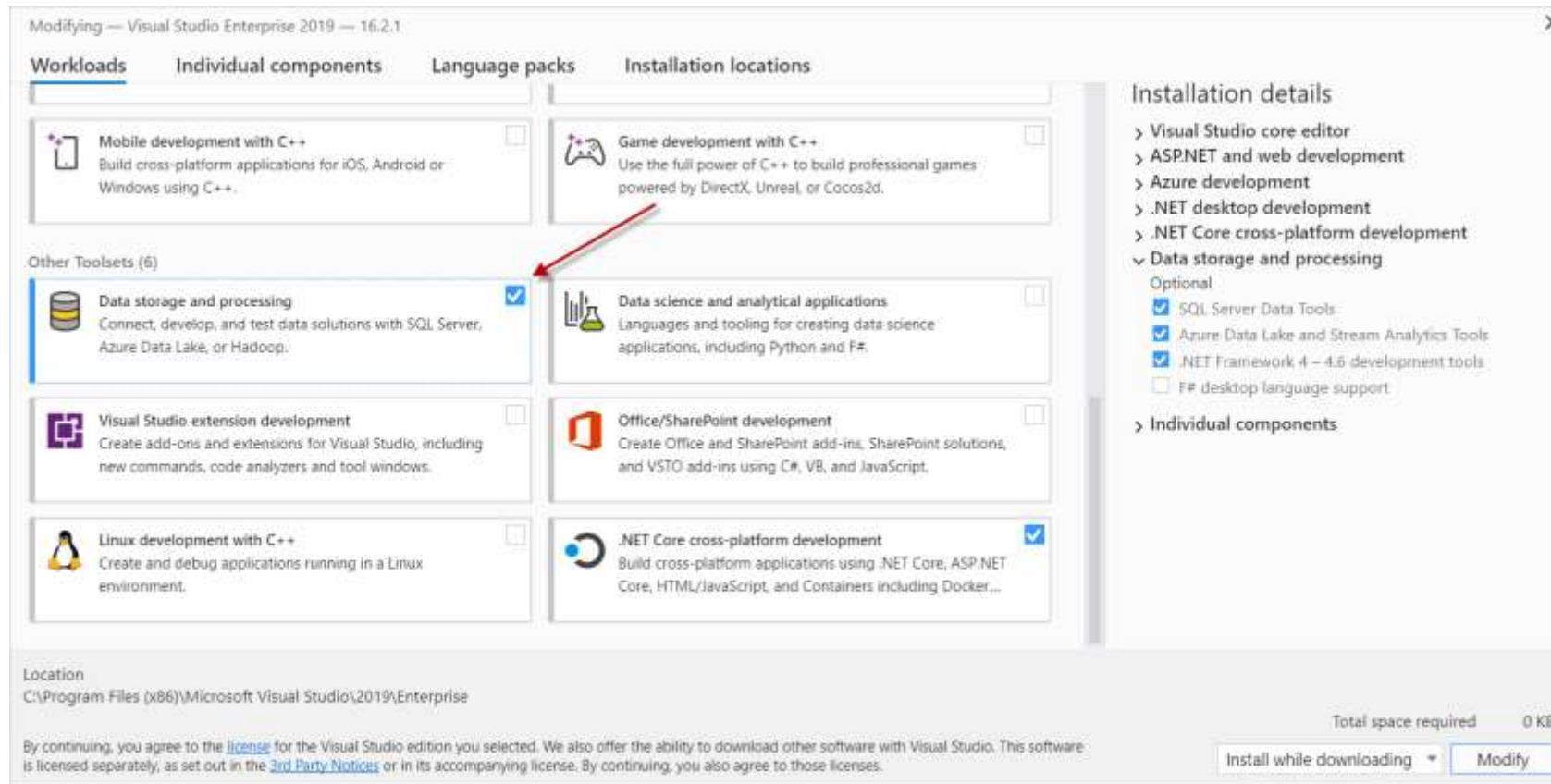
Aktuelle Versionen

Verwendete Software

- Visual Studio $\geq$ 2017
 - Alle Editionen
 - SQL Server Data Tools (SSDT)
- SQL Server
 - On-Premises, Virtuelle Maschine oder Docker Image ($\geq$ 2012)
 - SQL Azure
- SQL Server Management Studio 17.x/ 18.x

SQL Server Data Tools (SSDT) installieren

Installation via Visual Studio Installer (“Data storage and processing”)





Übersicht über die Optionen

Alle Änderungen von Hand protokollieren

- Excel, Word, ...
- Order mit T-SQL Skripten





Hier kein Thema

Datenbank-Backup

- (Ziel-) Struktur mit Daten
 - Datenbank wird neu erstellt/ überschrieben
 - Restore nur auf gleicher/ höherer Version möglich

Datenbank-Projekte

- Pflege & Entwicklung von Datenbanken
- Deployment direkt aus Visual Studio heraus

Schema & Data Compare

- Erstellung von „Diff-(T-SQL)-Skripte“ via Visual Studio 2017/ 2019
- Deployment per
 - Visual Studio 2017/ 2019

Data Tier Applications

- Erstellung von DACPAC & BACPAC via
 - Visual Studio 2017/ 2019
 - SQL Server Management Studio
- Importieren/ Deployen via
 - SQL Server Management Studio
 - Custom Code
 - ...



Datenbank-Projekte

Einführung

- „Quellcode“ einer Datenbank
 - Nur Schema, kein Inhalt
 - Design-Werkzeug
 - Datenbank-Projekte als Teil einer Solution
- CLR-Integration
 - Prozeduren & Funktionen mit .NET (C# & VB.NET)

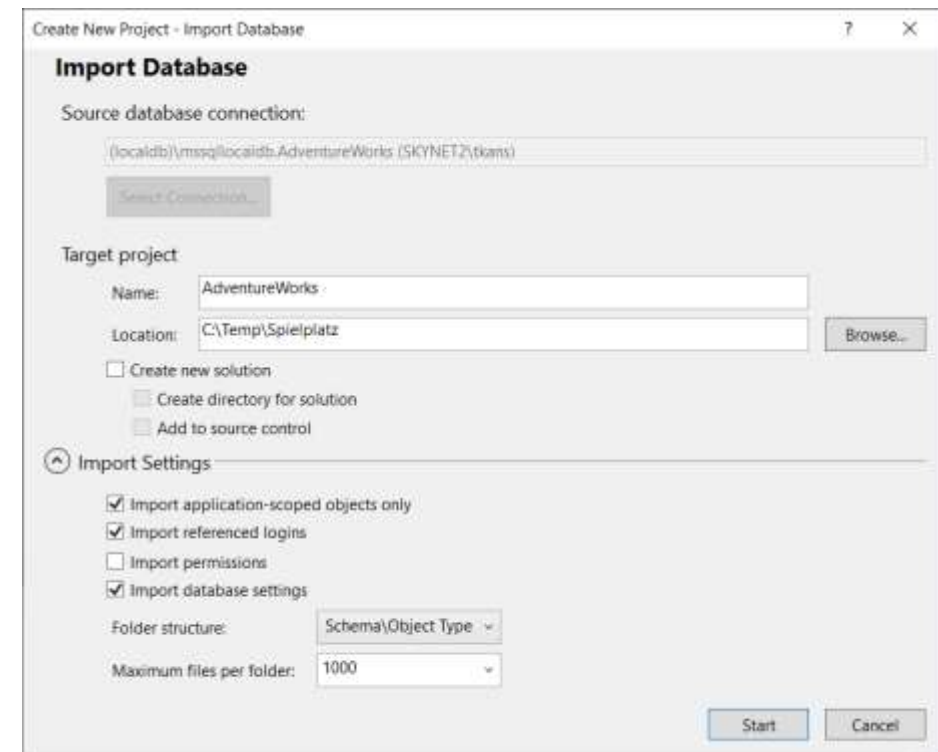
In Praxis-Projekten sollte das Datenbank-Projekt **oder** die Datenbank führend bei Änderungen sein

Datenbank-Projekte

- Erzeugen aus...
 - SQL Server Datenbank
 - SQL Server Datenbank mittels SQL Server Object Explorer
 - DACPAC (Data tier-Application)
 - SQL-Skript
- Alle Objekte sind „Create“-.sql-Dateien
 - VS „compiliert“ jedes Objekt => Fehler und Warnungen
 - ideal für Git, SVN & Co
 - *.jfm-Dateien ignorieren!
- Objekte können erzeugt, gelöscht oder modifiziert werden

Datenbank-Projekt anlegen

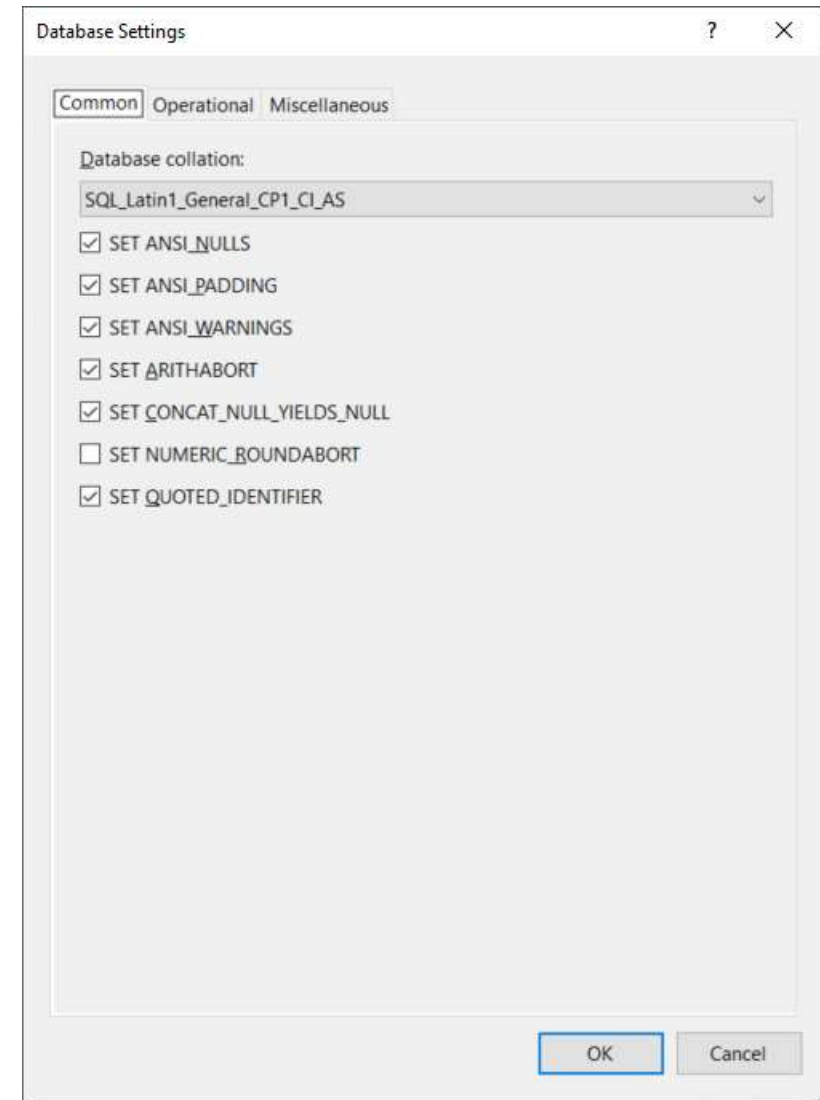
- Importieren aus
 - Bestehende Datenbank
 - Data-Tier Application (DACPAC)
 - Skript



Demo

Einstellungen

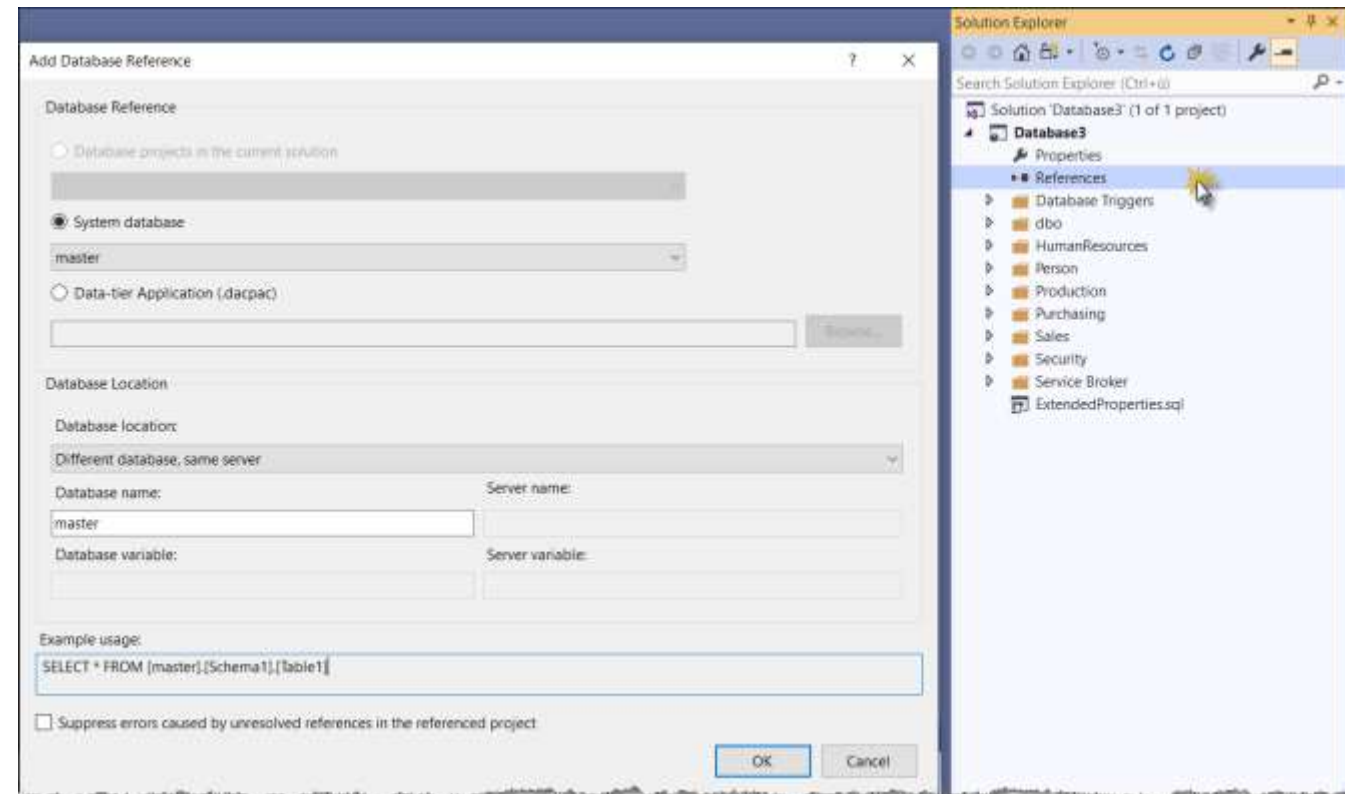
- Projekteinstellungen
- Data-tier Application
- Database Settings



Demo

Externe Referenzen

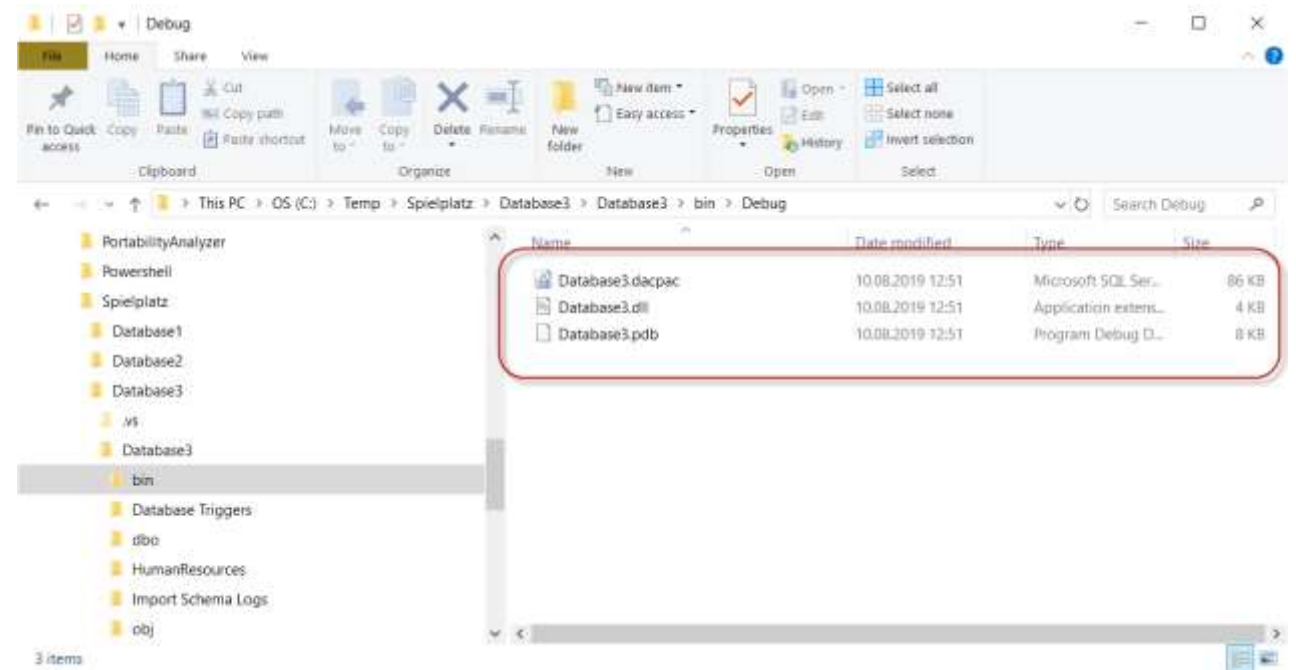
- Referenzen auf weitere Datenbanken
 - System-Datenbanken
 - Data-tier Application (DACPAC)
 - (andere) Datenbank-Projekte



Demo

Unter der Haube

- Kompilieren
 - Data-tier Application (DACPAC)
 - Optional: SQL Skript
 - DDL & PDB für CLR-Integration



Demo

A photograph of three bronze statues of chimpanzees sitting on a wooden park bench. The statue on the left is holding its hands to its ears. The middle statue is holding its hands to its face in a distressed or thoughtful pose. The statue on the right is holding its hands to its mouth. They are all wearing dark clothing and boots. The background features large evergreen trees and a building with a decorative wrought-iron gate. A semi-transparent dark blue banner with white text is overlaid across the middle of the image.

Schema & Data Compare

Schema Compare

- Vergleich der Struktur zweier Datenbanken
 - Datenbank
 - Datenbank-Projekt
 - Data-tier Application (DACPAC)
- Ziel anpassen
- Erzeugung von Delta-Skripten
- *.scmp-Dateien

Demo

Data Compare

- Inhaltsvergleich zwischen zwei Datenbanken
- Primary Key muss vorhanden sein
- Ziel anpassen
- Erzeugung von Delta-Skripten
- *.dcmp-Dateien

Demo

A close-up photograph of a bright yellow sunflower with a detailed brown and orange center. The flower is set against a background of green leaves and other sunflowers. A semi-transparent blue horizontal band is overlaid across the middle of the image, containing the text 'Data-tier Applications' in white. A solid dark blue horizontal bar is located at the bottom of the image.

Data-tier Applications

Data-tier Applications

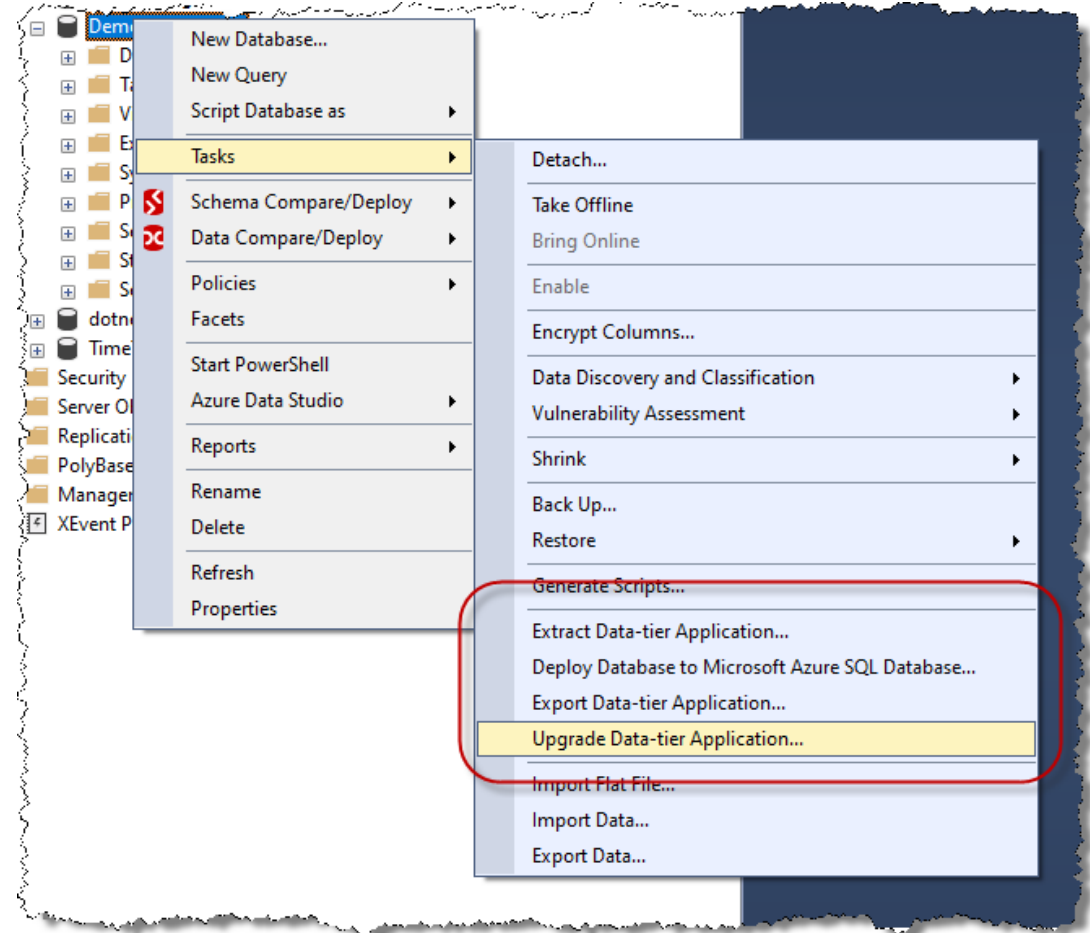
- DACPAC-Dateien
 - Nur Struktur
- BACPAC-Dateien
 - Struktur mit Daten
 - Export nach SQL Azure

DACPAC-Dateien

- (Ziel-) Struktur ohne Daten
 - Führt nur Anpassungen durch wo nötig
- Ein kompiliertes Datenbank-Projekt
 - Snapshot aus Database-Projekt
- Deploy/ Upgrade & Extract via
 - SQL Server Management Studio (SSMS)
 - .NET (Core)
 - Powershell
 - ...

Deployment mittels SSMS

- Deploy: DACPAC einspielen
- Upgrade: DACPAC aktualisieren
- Extract: DACPAC erstellen

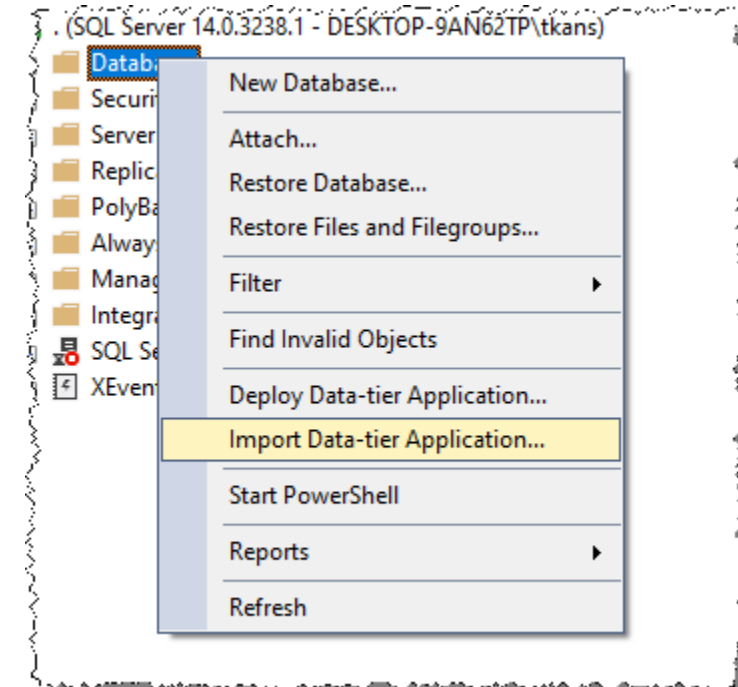
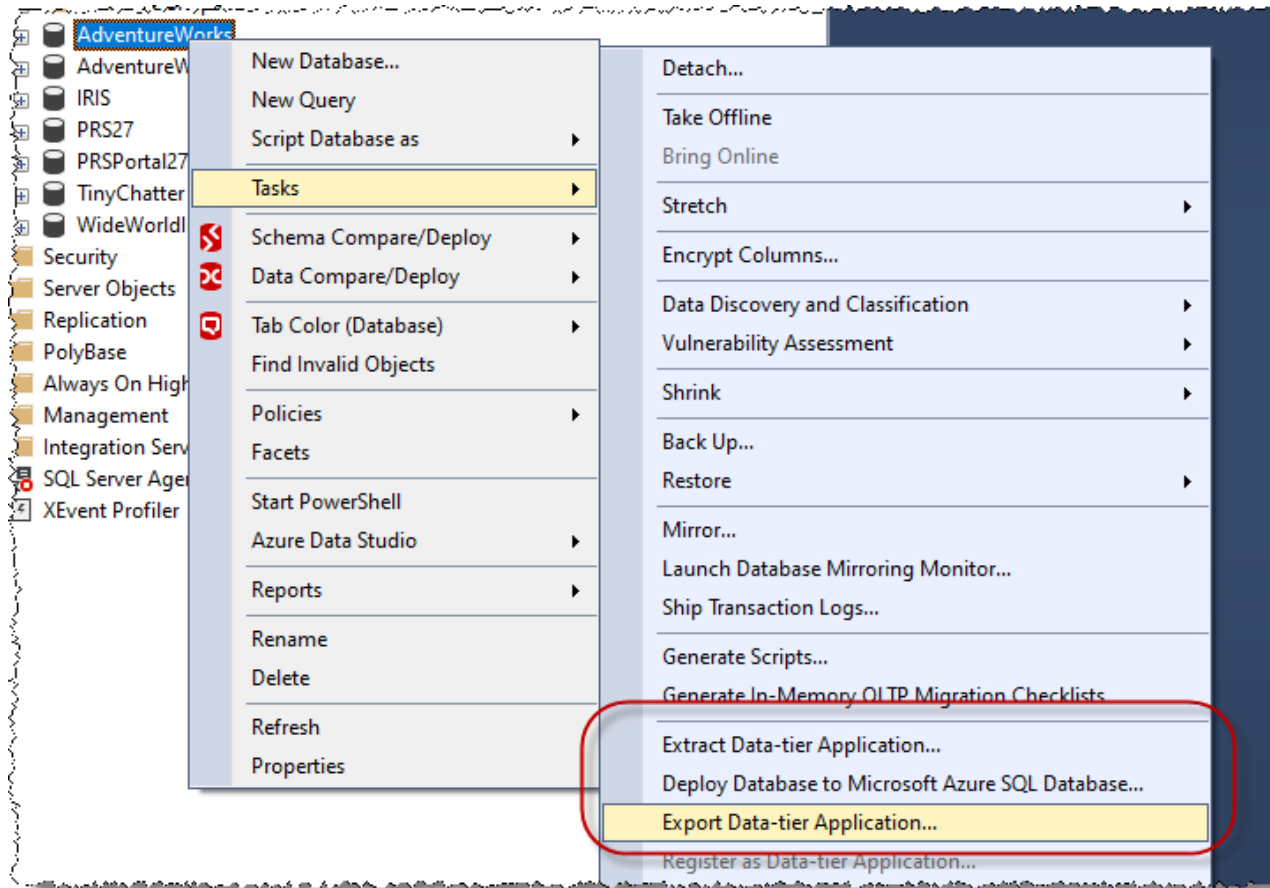


Demo

BACPAC-Dateien

- (Ziel-) Struktur mit Daten
 - Arbeitet nur mit leeren Datenbanken
 - Quelle must strukturelle fehlerfrei sein(!)
- Import & Export
 - SQL Server Management Studio (SSMS)
 - .NET (Core)
 - PowerShell
 - ...
- Datenbank Deployment nach SQL Azure möglich

BACPAC: Export & Import



Demo

DACPAC via .NET

```
private static void deployDACPAC()
{
    // DacService Instanz erzeugen und mit Zielserver verbinden
    var dacService = new DacServices("server = " + serverName);

    // DACPAC laden
    var dacpac = DacPackage.Load(dacpacPath);

    // Update, wenn die Datenbank bereits existiert.
    bool updateDatabase = existsDatabase();

    // Deployment ausführen
    dacService.Deploy(dacpac, targetDbName, updateDatabase);
}
```

Install-Package Microsoft.SqlServer.SqlManagementObjects

BACPAC via .NET

```
private static void importBACPAC()  
{  
    // DacService Instanz erzeugen und mit Zielserver verbinden  
    var dacService = new DacServices("server = " + serverName);  
  
    // BACPAC laden  
    var bacpac = BacPackage.Load(bacpacPath);  
  
    // Import ausführen  
    dacService.ImportBacpac(bacpac, targetDbName);  
}
```

```
\Program Files (x86)\Microsoft SQL Server Management Studio  
18\Common7\IDE\Extensions\Application\Microsoft.SqlServer.Dac.dll
```

Demo

Fragen?

Links



www.dotnetconsulting.eu



[@Tkansy](https://twitter.com/Tkansy)



tkansy@dotnetconsulting.eu