

SQL Server Data Tools (SSDT)

RNDr. David Gešvindr

MVP: Azure | MCSE: Data Platform | MCSD: Windows Store | MCT

david@wug.cz

 @gesvindr

Osnova

1. Představení nástroje SQL Server Data Tools
2. Vývoj databáze přímo na SQL Serveru
3. Vývoj databázového projektu bez připojení k databázi
4. Příprava databáze na týmový vývoj
5. Nasazení Continuous Integration
6. Databázové unit testy

Osnova

- 1. Představení nástroje SQL Server Data Tools**
2. Vývoj databáze přímo na SQL Serveru
3. Vývoj databázového projektu bez připojení k databázi
4. Příprava databáze na týmový vývoj
5. Nasazení Continuous Integration
6. Databázové unit testy

Představení nástroje SQL Server Data Tools

- Nástroj na vývoj databázových řešení, který se integruje do Visual Studio 2015
 - Od verze 2015 je součástí instalátoru
 - Zdarma ke stažení
- Hlavní scénáře použití:
 - Vývoj databázového projektu bez připojení k databázi
 - Vývoj databáze přímo na SQL Serveru
 - ◆ Překryv funkcionality se SQL Server Management Studiem
 - Sada nástrojů pro porovnání schématu i dat v databázi
 - Vývoj SQL CLR objektů
 - Vývoj Business Intelligence projektů (dříve Business Intelligence Development Studio)

Databázový model

- SSDT nepracují přímo s databází, ale udržují si její schéma v podobě deklarativního modelu
- Model je generován
 - z existující databáze
 - z databázového projektu, který je kolekcí SQL skriptů definujících objekty v databázi
- Tento model je využíván pro:
 - Implementaci databáze, změny databázových objektů, pokročilý refaktoring
 - Poskytuje IntelliSense, na který jsme zvyklí z Visual Studia
 - Je možné jej zkompilevat do podoby **SQL skriptu** nebo **DACPAC** balíčku
 - Konverzi databáze do více různých verzí SQL Serveru

Data Tier Application Framework

- **DACPAC**

- Schéma databáze a související objekty v XML
- Zjednodušení nasazení na SQL Server
- Automatický upgrade schématu
- Nepodporuje data

- **BACPAC**

- Schéma databáze v XML
- Data vyexportovaná s pomocí BCP
- Používá se pro migraci databáze do/z Azure SQL Database

Osnova

1. Představení nástroje SQL Server Data Tools
- 2. Vývoj databáze přímo na SQL Serveru**
3. Vývoj databázového projektu bez připojení k databázi
4. Příprava databáze na týmový vývoj
5. Nasazení Continuous Integration
6. Databázové unit testy

Vývoj databáze přímo na SQL Serveru

- Velmi podobná funkcionality jako nabízí **SQL Server Management Studio**
- **SQL Server Object Explorer** zpřístupňuje objekty v databázi
- Objekty je možné editovat s pomocí Designeru nebo v podobě SQL skriptu
- **Power Buffer**
 - Sleduje všechny změny, které provádíme nad otevřenými objekty v připojené databázi
 - Umožňuje vygenerovat **Change Skript** pro všechny změny najednou

Osnova

1. Představení nástroje SQL Server Data Tools
2. Vývoj databáze přímo na SQL Serveru
- 3. Vývoj databázového projektu bez připojení k databázi**
4. Příprava databáze na týmový vývoj
5. Nasazení Continuous Integration
6. Databázové unit testy

Vývoj databázového projektu

- SSDT přidávají do Visual Studia **SQL Server Database Project**
- Projekt uchovává schéma celé databáze v podobě sady DDL SQL skriptů
 - Každý objekt má svůj oddělený soubor s definicí
 - Výhodné kvůli zámkům a konfliktům při týmovém vývoji
- Virtuálně je nad těmito skripty vygenerován databázový model
 - IntelliSense, validace...
- V kombinaci se source controlem je verzována každá změna ve schématu databáze

Synchronizace změn mezi projektem a databází

- Nástroj **Schema Compare** umožní velmi rychle identifikovat a přenést změny mezi:
 - Databází a databázovým projektem
 - Dvěma databázemi
- Změny určené k přenosu jsou vygenerovány do podoby **change skriptu**, který můžete dále upravovat, rozšířit o migrace dat, otestovat a použít na produkci
- Nástroj hlídá, aby nedošlo k poškození uložených dat (lze vypnout)

Osnova

1. Představení nástroje SQL Server Data Tools
2. Vývoj databáze přímo na SQL Serveru
3. Vývoj databázového projektu bez připojení k databázi
- 4. Příprava databáze na týmový vývoj**
5. Nasazení Continuous Integration
6. Databázové unit testy

Příprava databáze na týmový vývoj

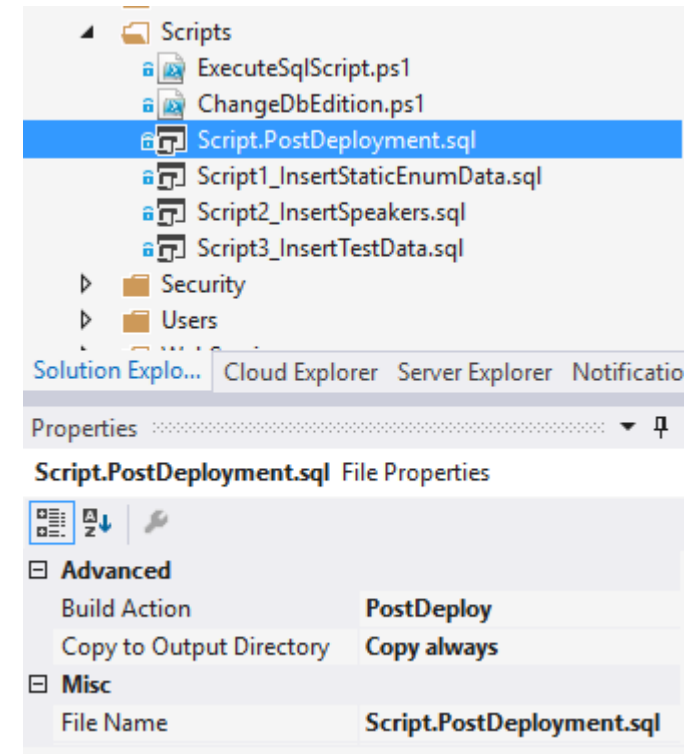
- Cíl: *Vývojář si stáhne ze source controlu aktuální verzi databáze, nasadí ji lokálně a může vyvíjet aplikaci*
- Schéma databáze je uloženo v databázovém projektu
- Je třeba vyřešit přenos dat
 - Číselníky, které vyžaduje aplikace pro svůj běh
 - Mnohdy není žádoucí vývoj aplikace nad „prázdnou“ databází, proto musíme připravit i testovací data jako součást projektu

Příprava databáze na týmový vývoj

1. Vytvoříme databázový projekt
2. Projekt nebo celé solution z Visual Studia publikujeme do source controlu
3. Import dat vyřešíme s pomocí SQL skriptů, které spustíme jako post-build akci
4. Vytvoříme deployment konfiguraci pro nasazení databáze

Příprava skriptu vkládajícího data

- Pro vygenerování skriptu na vložení dat je možné použít nástroj **Data Compare**
- S pomocí něj se nechá vygenerovat SQL skript na vložení dat do prázdné databáze
- Skript je nutné označit build akcí **PostDeploy**, aby se sám spustil po nasazení databáze

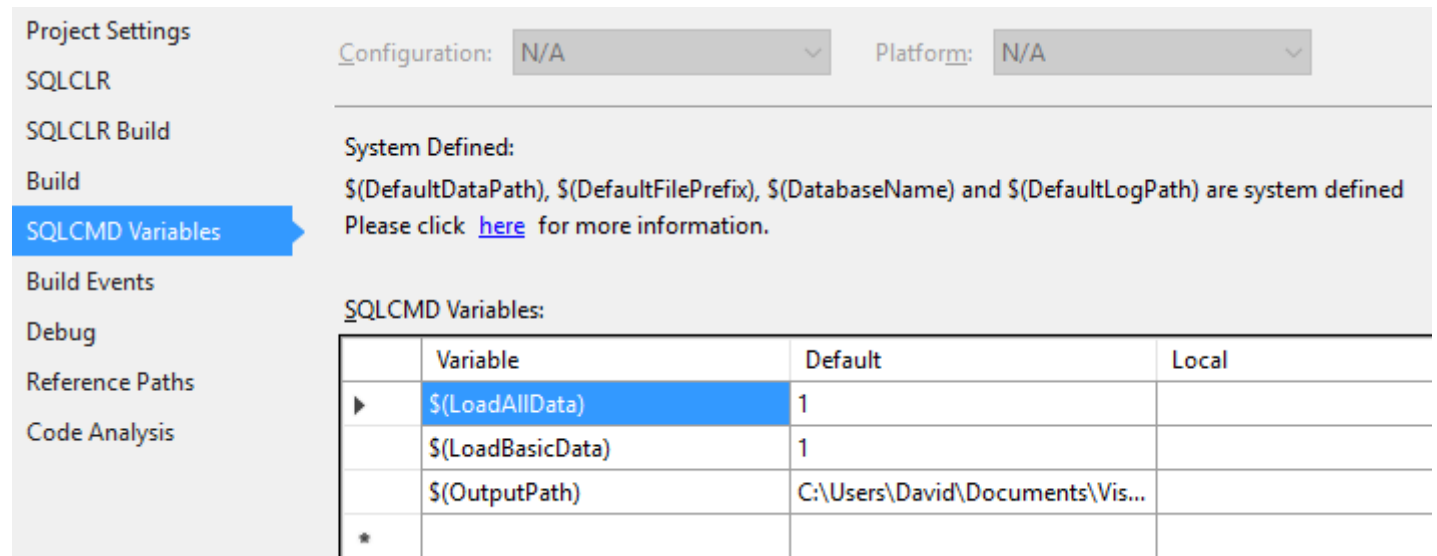


Parametrizace skriptů

- PostDeploy skripty mohou obsahovat vstupní parametry
 - Lze jimi ovlivnit např. jaká část dat bude nakopírována do databáze
- **SQLCMD Variables** je nutné definovat v nastavení projektu

```
IF ($(LoadBasicData) = 1)
BEGIN
    PRINT 'Post-build load basic enum data'
    :r ".\Script1_InsertStaticEnumData.sql"
    :r ".\Script2_InsertSpeakers.sql"
    :r ".\Script3_InsertTestData.sql"

    IF ($(LoadAllData) = 1)
    BEGIN
        PRINT 'Post-build load all data'
    END
END
ELSE BEGIN
    PRINT 'Post-build data load skipped'
END
```



The screenshot shows the 'SQLCMD Variables' tab in the Visual Studio Project Settings dialog. The left sidebar lists various settings categories, with 'SQLCMD Variables' selected and highlighted in blue. The main area shows the configuration for the selected variables. At the top, there are dropdown menus for 'Configuration' and 'Platform', both set to 'N/A'. Below this, a section titled 'System Defined:' lists variables like \$(DefaultDataPath), \$(DefaultFilePrefix), \$(DatabaseName), and \$(DefaultLogPath), with a note that they are system defined and a link to 'here' for more information. The 'SQLCMD Variables:' section contains a table with columns for Variable, Default, and Local. The table lists three variables: \$(LoadAllData) with a default of 1, \$(LoadBasicData) with a default of 1, and \$(OutputPath) with a default of C:\Users\David\Documents\Vis... The first row is highlighted in blue.

	Variable	Default	Local
▶	\$(LoadAllData)	1	
	\$(LoadBasicData)	1	
	\$(OutputPath)	C:\Users\David\Documents\Vis...	
*			

Osnova

1. Představení nástroje SQL Server Data Tools
2. Vývoj databáze přímo na SQL Serveru
3. Vývoj databázového projektu bez připojení k databázi
4. Příprava databáze na týmový vývoj
- 5. Nasazení Continuous Integration**
6. Databázové unit testy

Nasazení Continuous Integration

- Kompilace databáze a její nasazení bude realizováno build serverem na základě aktuálních zdrojových kódu v source controlu
- Výhody:
 - Automatické ověření, že uložený projekt v source controlu je zkompilovatelný
 - Automatická publikace databáze pro staging verzi aplikace po každé změně databáze
 - Automatické pouštění databázových unit testů po každé změně databáze

Konfigurace build serveru

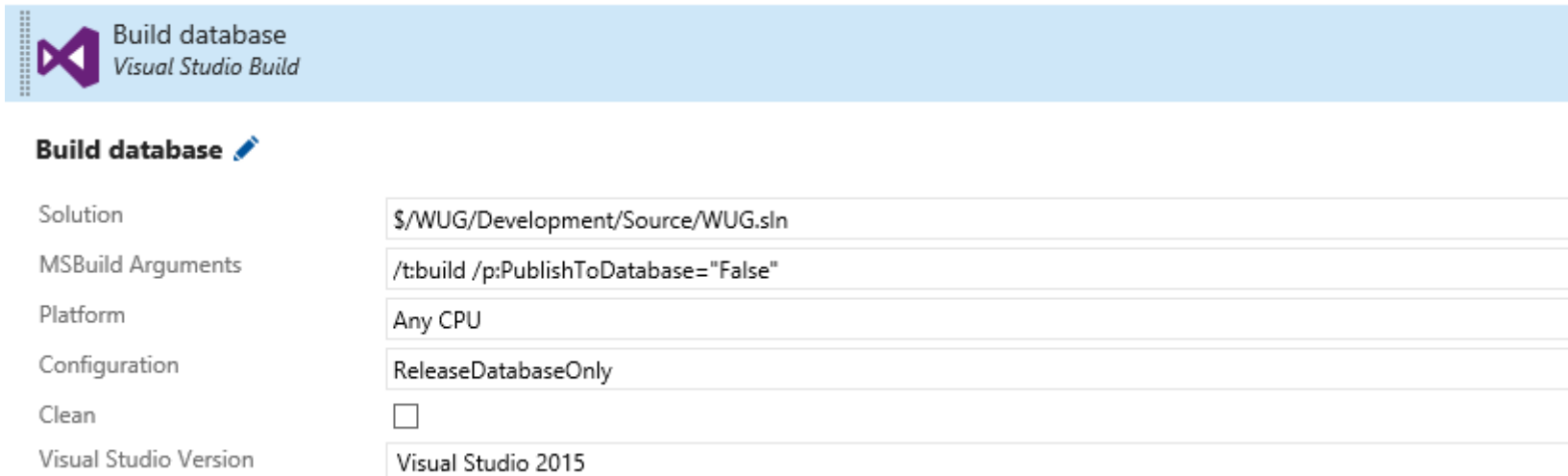
- Databázový projekt se kompiluje přes MSBuild jako jiné projekty
- Z tohoto důvodu je možné využívat jakýkoliv build server

- Visual Studio Online
 - Cloudová varianta Team Foundation Serveru
 - Zahrnuje zdarma 240 minut běhu buildů
 - Ve verzi 2015 obsahuje nový build server

- V rámci daného solution vytvoříme build konfiguraci, která kompiluje jen databázový projekt

Kompilace databázového projektu

- Nejprve necháme zkompileovat databázový projekt

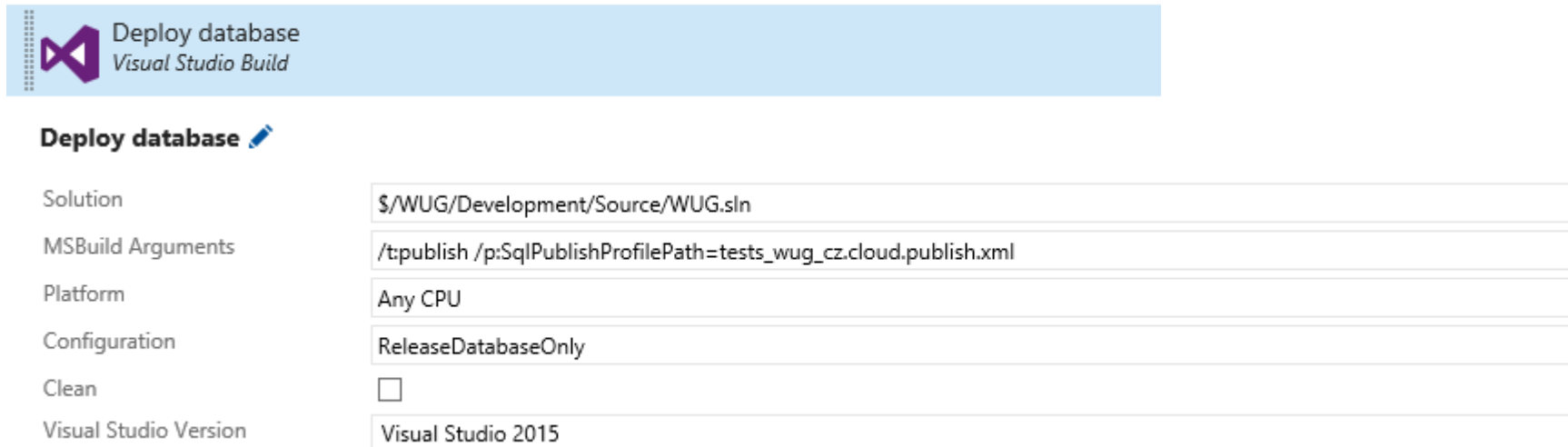


The screenshot shows the configuration for a build task named 'Build database' in Visual Studio Build. The configuration is displayed in a light blue header bar with the Visual Studio logo and the text 'Build database Visual Studio Build'. Below the header, the task name 'Build database' is followed by a pencil icon. The configuration is presented as a list of key-value pairs:

Solution	\$/WUG/Development/Source/WUG.sln
MSBuild Arguments	/t:build /p:PublishToDatabase="False"
Platform	Any CPU
Configuration	ReleaseDatabaseOnly
Clean	<input type="checkbox"/>
Visual Studio Version	Visual Studio 2015

Nasazení databázového projektu

- Po zkompilování databázového projektu jej necháme nasadit na SQL Server nebo do Azure SQL Database



The screenshot shows the configuration for the 'Deploy database' task in Visual Studio Build. The task is titled 'Deploy database' with a blue pencil icon. The configuration is as follows:

Property	Value
Solution	\$/WUG/Development/Source/WUG.sln
MSBuild Arguments	/t:publish /p:SqlPublishProfilePath=tests_wug_cz.cloud.publish.xml
Platform	Any CPU
Configuration	ReleaseDatabaseOnly
Clean	<input type="checkbox"/>
Visual Studio Version	Visual Studio 2015

- Umístění cílové databáze je dáno publikačním profilem, který jsme založili v našem projektu

Osnova

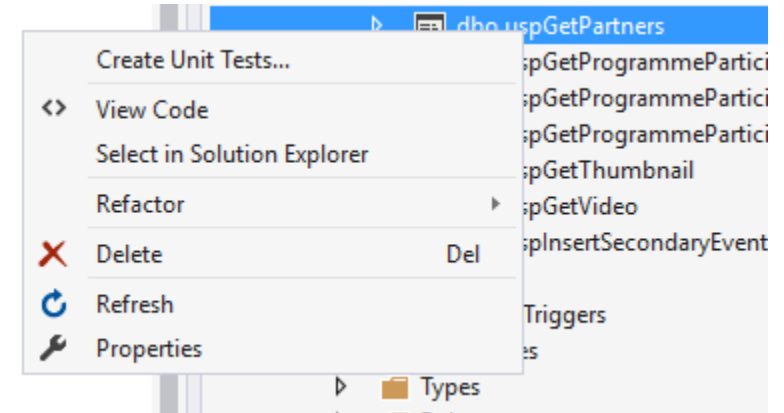
1. Představení nástroje SQL Server Data Tools
2. Vývoj databáze přímo na SQL Serveru
3. Vývoj databázového projektu bez připojení k databázi
4. Příprava databáze na týmový vývoj
5. Nasazení Continuous Integration
- 6. Databázové unit testy**

Databázové unit testy

- Opakovaně ručně testovat uložené procedury, funkce a pohledy je velmi časově náročné
 - Je nutné připravit data a ověřit, že výsledky odpovídají očekávání
- Tento problém řeší automatizace databázových testů
 - Databázový unit test automaticky ověří korektnost testovaného objektu
- Problém: Unit testy vyžadují, aby v databázi byla „známá“ data
 - Je třeba rozhodnout, která data budou součástí build skriptů databáze a která data si nachystá přímo test
 - Po provedení testu je třeba **efektivně uklidit**, aby se testy neovlivňovaly

Podpora databázových testů v SSDT

- Ze SQL Server Object Exploreru je možné si nechat vygenerovat databázový unit test
- Databázový unit test se skládá z:
 - Pre-test – příprava data pro test
 - Test – exekuce testovaného objektu, porovnání výsledků
 - Post-test – úklid
- Je několik způsobů, jak ověřit správnost výsledku:
 - Počet řádků, které se vrací po exekuci
 - Návratová hodnota
 - Data Checksum – kontrola, že se vrací identické výsledky



Vlastní implementace

- Dostupné databázové unit testy nebyly dostatečně flexibilní
 - Nedostatek možností, jak jednoduše validovat výsledky
 - Zdlouhavá tvorba
- Pro testování databáze používáme klasický testovací projekt a unit testy
 - Využíváme TransactionScope pro izolaci všech změn provedených při běhu testu, které jsou po dokončení testu zrušeny (roll-back)
 - Pro komunikaci s databází využíváme O/R mapper LINQ to SQL
 - ◆ Voláme s pomocí něj testované objekty
 - ◆ Nastavujeme a kontrolujeme data v databázi

Osnova

1. Představení nástroje SQL Server Data Tools
2. Vývoj databáze přímo na SQL Serveru
3. Vývoj databázového projektu bez připojení k databázi
4. Příprava databáze na týmový vývoj
5. Nasazení Continuous Integration
6. Databázové unit testy