



SQL Server Data Tools

SNEK 4 – April 2015
Bernd Jungbluth

www.berndjungbluth.de

■ Zur Person

■ Bernd Jungbluth

■ IT-Erfahrung

- SQL und Datenbanken seit 1991
- SQL Server seit Version 7.0

■ Freiberuflicher Berater und Entwickler

- Administration SQL Server
- Entwicklung und Optimierung von SQL Server-Datenbanken
- Datawarehouse-Systeme nach Bill Inmon
- SQL Server Reporting Services
- SQL Server Integration Services
- Migration Access nach SQL Server

■ Agenda

■ SQL Server Data Tools

- Inhalt
- Versionen

■ SQL Server Data Tools für Datenbankprojekte

- Online-Entwicklung
- Offline-Entwicklung

■ Datenbankprojekte

- Entwickeln
- Bereitstellen

■ Zusammenfassung und Fazit

■ SQL Server Data Tools / Inhalt

■ SQL Server Data Tools

- Entwicklungsumgebung in Visual Studio
- Für Datenbankprojekte
- Für Business Intelligence Projekte – SSIS, SSAS, SSRS

■ Verfügbarkeit

- Enthalten in SQL Server 2012
- SSDT für BI-Projekte – SSIS, SSAS, SSRS
- Nachinstallation von SSDT für Datenbankprojekte erforderlich
- Beides nicht mehr enthalten in SQL Server 2014
- Als eigenständiger Download für Visual Studio verfügbar
- SSDT für BI-Projekte – SSIS, SSAS, SSRS
- SSDT für Datenbankprojekte
- Von SQL Server und Visual Studio unabhängige Weiterentwicklung

■ SQL Server Data Tools / Versionen

■ Versionen

- Abhängig von der installierten Version von Visual Studio
 - VS 2010, 2012, 2013, vNext – in der jeweiligen Sprachversion
- Abhängig von der Version des SQL Servers
 - SSDT für die jeweils neueste Version von SQL Server
 - SSDT-BI zum Teil für die jeweils neueste Version von SQL Server

■ Downloads

- SSDT für Visual Studio 2012
- SSDT-BI für Visual Studio 2012
- SSDT-BI für Visual Studio 2013
 - Inklusive entsprechender Visual Studio Shell
- SSDT für Visual Studio 2013 nur mit Visual Studio 2013 verfügbar
 - Auch in kostenlosen Express-Versionen enthalten

■ Agenda

■ SQL Server Data Tools

- Inhalt
- Versionen

■ SQL Server Data Tools für Datenbankprojekte

- Online-Entwicklung
- Offline-Entwicklung

■ Datenbankprojekte

- Entwickeln
- Bereitstellen

■ Zusammenfassung und Fazit

■ SQL Server Data Tools für Datenbankprojekte

■ SQL Server Data Tools für Datenbankprojekte

- Datenbankentwicklung als Bestandteil von ALM
- ALM = »Application Lifecycle Management«

■ Application Lifecycle Management

- Applikationsentwicklung
- Anforderungen, Entwurf und Entwicklung
- Applikationsbetrieb
- Bereitstellung, Betrieb und Weiterentwicklung

■ SQL Server Data Tools und ALM

- Entwickeln
- Testen
- Bereitstellen

■ SQL Server Data Tools für Datenbankprojekte

■ SQL Server Data Tools und SQL Server

- Datenbankentwicklung für SQL Server
- SQL Server in allen Editionen ab Version 2005
- SQL Azure

■ SQL Server Data Tools und Visual Studio

- Projektvorlage in Visual Studio
- »SQL Server-Datenbankprojekt«
- Primär skriptbasierte Entwicklung
- Graphische Unterstützung lediglich beim Tabellenentwurf
- Bekannte Entwicklungsunterstützung von Visual Studio
- IntelliSense, Debugging, Refactoring, Unit-Testing, etc.
- Entwicklungsunterstützung abhängig vom Entwicklungsmodus
- Online-Entwicklung und Offline-Entwicklung

■ SQL Server Data Tools für Datenbankprojekte / **Online-Entwicklung**

■ Online-Entwicklung

- Ähnlich der Entwicklung im SQL Server Management Studio
- Verbindung zur SQL Server-Datenbank
- Direktes Speichern der Änderungen in SQL Server-Datenbank
- Keine projektbasierte Entwicklung
- Nur manuelles Skripten von Datenbankobjekten möglich
- Speichern der Skripte als Dateien im Dateisystem

■ Möglichkeiten

- Erstellen und Ändern von Datenbankobjekten
- Tabellen, Gespeicherte Prozeduren, Sichten, Funktionen, etc.
- So gut wie keine Entwicklungsunterstützung von Visual Studio
- Kein Refactoring, kein Unit-Testing, kein Erstellen und Bereitstellen
- Lediglich Debugging verfügbar

■ SQL Server Data Tools für Datenbankprojekte / **Offline-Entwicklung**

■ Offline-Entwicklung

- Keine Verbindung zur SQL Server-Datenbank
- Projektbasierte Datenbankentwicklung
 - Speichert Datenbankobjekte als Skriptdateien
- Import bestehender Datenbanken möglich

■ Projektbasierte Datenbankentwicklung

- Erstellen und Ändern von Datenbankobjekten – als Skripte
 - Tabellen mit deren Datenintegrität, Indizes und Trigger
 - Gespeicherte Prozeduren, Sichten und Funktionen
 - Benutzerrechte, SQL Skripte mit Batch-Anweisungen, etc.
- Skriptübergreifende Prüfung von Abhängigkeiten und T-SQL-Syntax
 - »SQL Server T-SQL Compiler Service«
- Erstellen und Bereitstellen des Datenbankprojekts

■ Agenda

■ SQL Server Data Tools

- Inhalt
- Versionen

■ SQL Server Data Tools für Datenbankprojekte

- Online-Entwicklung
- Offline-Entwicklung

■ Datenbankprojekte

- Entwickeln
- Bereitstellen

■ Zusammenfassung und Fazit

■ Datenbankprojekte

■ Datenbankprojekt

- Sammlung der Skripte zu den einzelnen Datenbankobjekten
- Einheitliche Konfiguration zum Entwickeln, Testen und Bereitstellen
- Definition von Codierungsregeln

■ Projekt-Snapshot

- Speichert aktuellen Stand des Datenbankprojekts
- Synchronisation von Projekt-Snapshot zu Datenbankprojekt möglich

■ Quellcodeverwaltung

- Zentrale Verwaltung des Quellcodes
- »Team-Foundation-Versionskontrolle« und »GIT«
- Speicherort Lokal, im Netzwerk und in der Cloud
- Unterstützt bekannte Funktionen wie Änderungshistorie

■ Datenbankprojekte

■ Demo

- Erstellen eines Datenbankprojekts
- Konfiguration der Quellcodeverwaltung
- Einrichten der Entwicklungsumgebung
- Konfiguration des Datenbankprojekts
- Erstellen eines Projekt-Snapshots

■ Datenbankprojekte / **Entwickeln**■ **Datenbankentwicklung**

- Erstellen einzelner Datenbankobjekte als Skripte
 - Speichern der Skripte in frei definierbarer Ordnerstruktur
- Entwicklung in T-SQL
 - Lediglich für Tabellen graphische Oberfläche verfügbar
- Unterstützt SQL Server ab Version 2005
- Import bestehender Datenbanken möglich

■ **Import von Datenbanken**

- Erstellt pro Objekt der Datenbank ein Skript
- Verschiedene Importmöglichkeiten
 - SQL-Skript
 - dacpac-Datei
 - Direkter Datenbankzugriff

■ Datenbankprojekte / **Entwickeln**

■ **Demo**

- Import einer Datenbank
- Ändern von Datenbankobjekten
- Erstellen von Datenbankobjekten

■ Datenbankprojekte / Entwickeln / Datenbankentwicklung mit Daten

■ Datenbankentwicklung mit Daten

- Entwicklung ohne Daten mühsam bis unmöglich
- Gespeicherte Prozeduren, Sichten, Funktionen und Trigger
- Entwicklerdatenbank mit Daten sinnvoll
- »Debugdatenbank«

■ SQL Server LocalDB

- SQL Server Express Edition für Entwickler
- Applikationsbezogene Instanz im Kontext des Benutzers
- Prozess unter der Anmeldung des aktuellen Benutzers
- Instanz für andere Entwickler nicht verfügbar
- Instanzname abhängig von SQL Server-Version im Datenbankprojekt
- Beispiel für Projekte zu SQL Server 2014: *(localdb)\projectsV12*
- Administration mit SSDT

■ Datenbankprojekte / Entwickeln / Debugdatenbank**■ Debugdatenbank**

- Konfiguration in den Eigenschaften des Datenbankprojekts
- Veröffentlichen des Datenbankprojekts in SQL Server LocalDB
- Erstellen und Bereitstellen des Datenbankprojekts

■ Daten

- Datenabgleich mit Produktivdatenbank
- Für die komplette Datenbank oder pro Tabelle
- Per Skript mit INSERT-Anweisungen
- Eingabe von Testdaten

■ Testdaten

- Generieren von Testdaten nicht unterstützt
- Laut SSDT-Entwicklerteam vorerst nicht geplant

■ Datenbankprojekte / Entwickeln / **Debugdatenbank**

■ Entwickeln in Debugdatenbank

- Abfragefenster ähnlich SQL Server Management Studio
- Verbindung, Ausgabe Ausführungsplan, Abfrageoptionen, etc.
- Ausführen von T-SQL per Tastenkombination *STRG+UMSCHALT+E*
- Vertraute Einstellung *F5* reserviert
- Veröffentlichen des Datenbankprojekts in SQL Server LocalDB
- Tastenkombination in Optionen konfigurierbar

■ Konfiguration F5

- Menüpunkt *Extras -> Optionen -> Umgebung -> Tastatur*
- SQL.TSqlEditorAbfrageAusführen* auswählen
- Microsoft SQL Server Data Tools, T-SQL-Editor* auswählen
- F5* drücken
- Schaltfläche *Zuweisen* klicken

■ Datenbankprojekte / Entwickeln / **Debugdatenbank**

■ Demo

- Verwalten von Instanzen der SQL Server LocalDB
- Veröffentlichen des Datenbankprojekts als Debugdatenbank
- Datenimport aus Produktivdatenbank
- Entwickeln mit Daten
- Konfiguration der Tastenkombination

■ Datenbankprojekte / Entwickeln / **Entwicklungsunterstützung**

■ **Entwicklungsunterstützung**

- Möglichkeiten und Funktionalität von Visual Studio
 - IntelliSense mit Vorschau
 - Gehe zu Definition
 - Abhängigkeitsanalyse
 - Refactoring
 - Debugging
 - Quellcode-Vergleich
 - Modultests
 - Codeanalyse

■ **Informationen**

- Ausgabe von Fehler, Warnungen und Informationen
- Auswirkungsanalyse beim Ändern von Datenbankobjekten

■ Datenbankprojekte / Entwickeln / **Entwicklungsunterstützung**

■ Refactoring

- Automatisches Erkennen und Durchführen von Änderungen
- Umbenennen von Bezeichnungen
- Ersetzen von Platzhaltern wie *SELECT* *

■ Debugging

- Gespeicherte Prozeduren, Funktionen, Trigger und SQL-Anweisungen
- Setzen von Haltepunkten, Ausgabe von Werten zur Laufzeit, etc.

■ Quellcode-Vergleich

- Vergleichen und Synchronisieren von Änderungen
- Schemavergleich vom Projekt-Snapshot zu Datenbankprojekt
- Schemavergleich von Debugdatenbank zu Datenbankprojekt
- Änderungshistorie und Gegenüberstellung in der Quellcodeverwaltung

■ Modultests

- Im Original »Unit-Testing«
- Automatisierte Tests
- Test und Ergebnisanalyse von Programmen durch Programme
- Eigenes .NET-Projekt im Datenbankprojekt
- Nur verfügbar ab Visual Studio Professional
- Test-Driven Development denkbar

■ Test-Unit

- Testeinheit zum Testen von Datenintegrität und Programmlogik
- Primärschlüssel, Einschränkungen, Fremdschlüssel, UNIQUE-Keys
- Sichten, Gespeicherte Prozeduren, Funktionen und Trigger
- Bietet unterschiedliche Standardprüfungen zur Ergebnisanalyse
- Definition von Vor- und Nachtests möglich

■ Datenbankprojekte / Entwickeln / **Entwicklungsunterstützung**

■ Demo

- Ersetzen von Platzhalten
- Umbenennen einer Spalte
- Debugging einer Tabellenwertfunktion
- Schemavergleich mit Projekt-Snapshot

■ Datenbankprojekte / Entwickeln / Codeanalyse

■ Codeanalyse

- Analyse des Quellcodes anhand definierter Codierungsregeln
- Auswirkung als Warnung oder Fehler
- Optionale Auswirkung beim Erstellen des Datenbankprojekts
- Definition eigener Codierungsregeln mittels .NET-Entwicklung

■ Codierungsregeln

- Unterteilt in drei Kategorien
- Kategorie »Datenbankdesign«
 - Beispiel: Keine Verwendung von *SELECT* *
- Kategorie »Bezeichnungen«
 - Beispiel: Keine eigenen Gespeicherten Prozeduren mit Präfix *sp_*
- Kategorie »Performance«
 - Beispiel: Kein LIKE-Vergleich mit beginnendem %-Zeichen

■ Datenbankprojekte / Entwickeln / Codeanalyse

■ Demo

- Konfiguration der Codeanalyse
- Manuelle Codeanalyse
- Automatische Codeanalyse beim Erstellen

■ Agenda

■ SQL Server Data Tools

- Inhalt
- Versionen

■ SQL Server Data Tools für Datenbankprojekte

- Online-Entwicklung
- Offline-Entwicklung

■ Datenbankprojekte

- Entwickeln
- Bereitstellen

■ Zusammenfassung und Fazit

■ Datenbankprojekte / Bereitstellen / Erstellen, Debuggen und Bereitstellen**■ Erstellen eines Datenbankprojekts**

- Im Original »Build«
- Prüfen der T-SQL-Syntax
- Prüfen der Abhängigkeiten zwischen den Datenbankobjekten
- Liefert ausführliche Übersicht der Warnungen und Fehler
- Ohne fehlerfreies Erstellen kein Bereitstellen möglich

■ Debuggen eines Datenbankprojekts

- Erstellen des Datenbankprojekts
- Bereitstellen des Datenbankprojekts in SQL Server LocalDB

■ Bereitstellen eines Datenbankprojekts

- Erstellen des Datenbankprojekts
- Veröffentlichen des Datenbankprojekts in Ziel-Instanz

■ Datenbankprojekte / Bereitstellen

■ Bereitstellen

- Installation des Datenbankprojekts in Ziel-Instanz
- Neue Datenbank in Ziel-Instanz
- Anlegen und Ändern von Datenbankobjekten in Ziel-Datenbank
- Fehlerfreies Erstellen des Datenbankprojekts erforderlich
- Mehrere Möglichkeiten der Bereitstellung
- Veröffentlichen, Schemavergleich, SQL-Skript, dacpac-Datei

■ Veröffentlichen

- Bereitstellen des Datenbankprojekts in Ziel-Instanz
- Anlegen neuer Datenbank oder Ändern bestehender Datenbank
- Verbindung von SSDT zur Ziel-Instanz erforderlich
- Mehrere Optionen zum Veröffentlichen verfügbar
- Speichern der Verbindung und Konfiguration als Profil möglich

■ Datenbankprojekte / Bereitstellen / **Veröffentlichen**

■ Demo

Veröffentlichen des Datenbankprojekts

■ Datenbankprojekte / Bereitstellen / Schemavergleich und Installationskript

■ Schemavergleich

- Vergleich des Datenbankprojekts mit Ziel-Datenbank
- Graphische Gegenüberstellung der Unterschiede
- Verbindung von SSDT zu Ziel-Datenbank erforderlich
- Nicht geeignet zur Installation einer neuen Datenbank

■ SQL-Skript

- Bereitstellen des Datenbankprojekts in Ziel-Instanz
 - Enthält Befehle zum Anlegen einer Datenbank
 - Nicht geeignet zum Ändern einer bestehenden Datenbank
- Keine Verbindung von SSDT zu Ziel-Datenbank erforderlich
- Ausführen des Skripts in Ziel-Instanz nur per SQLCMD möglich
- Nach Erstellen des Datenbankprojekts im *bin*-Verzeichnis verfügbar
 - Name des Skripts: *Datenbankprojekt_Create.sql*

■ Datenbankprojekte / Bereitstellen / **Schemavergleich und Installationskript**

■ Demo

Bereitstellen per SQL-Skript

■ Datenbankprojekte / Bereitstellen / **dacpac-Datei**■ **dacpac-Datei**

- Datei zur Installation des Datenbankprojekts in Ziel-Instanz
- Anlegen neuer Datenbank oder Ändern bestehender Datenbank
- Beschreibt das Datenbankprojekt mit XML und SQL
- Konfiguration der dacpac-Datei in Projekteinstellungen
- Name der Datenbank, Versionsnummer und Beschreibung
- Nach Erstellen des Datenbankprojekts im *bin*-Verzeichnis verfügbar
- Keine Verbindung von SSDT zur Ziel-Instanz erforderlich

■ **Installation**

- Im SQL Server Management Studio der Ziel-Instanz
- Anlegen oder Aktualisieren einer »Datenebenenanwendung«
- Bietet Auswirkungsanalyse mit Darstellung möglicher Folgen
- Erlaubt ROLLBACK bei fehlerhafter Installation

■ Datenbankprojekte / Bereitstellen / Datenebenenanwendung**■ Datenebenenanwendung**

- Im Original »data-tier application« (DAC)
- Registrierung einer Datenbank als Datenebenenanwendung
 - Speichern der Metadaten im Repository
- Konfiguration über Kontextmenübefehl *Tasks* der Datenbank

■ Möglichkeiten

- Einfache Installation von Änderungen per dacpac-Datei
- Verwalten verschiedener Versionen einer Datenbank
- Exportieren als bacpac-Datei
 - Speichert die Datenbankobjekte ohne Daten
- Extrahieren als dacpac-Datei
 - Zum Übertragen der Datenebenenanwendung in eine andere Instanz
- Bereitstellen als Azure SQL-Datenbank und in Azure-VM

■ Datenbankprojekte / Bereitstellen / **Datenebenenanwendung und dacpac-Datei**

■ Demo

- Registrieren einer Datenbank als Datenebenenanwendung
- Bereitstellen des Datenbankprojekts per dacpac-Datei

■ Agenda

■ SQL Server Data Tools

- Inhalt
- Versionen

■ SQL Server Data Tools für Datenbankprojekte

- Online-Entwicklung
- Offline-Entwicklung

■ Datenbankprojekte

- Entwickeln
- Bereitstellen

■ Zusammenfassung und Fazit

■ Zusammenfassung und Fazit

■ SQL Server Data Tools

- SSDT für Datenbankprojekte
- SSDT-BI für SSIS, SSAS und SSRS

■ Versionen

- Mehrere Downloads für verschiedene Versionen
- Abhängig von Visual Studio und SQL Server
- Ganz nett ausgedrückt: **ES IST EIN KRAMPF!**

■ SQL Server Data Tools für Datenbankprojekte

- Projektbasierte Datenbankentwicklung
- Offline-Entwicklung von Datenbanken
- Hilfreiche Entwicklungsunterstützung
- Mehrere Möglichkeiten zum Bereitstellen

■ Quellen und Hinweise

■ Quellen

- SQL Server Data Tools
 - <https://msdn.microsoft.com/de-de/library/hh297027.aspx>
- Updates bei SQL Server Data Tools Team Blog
 - <http://blogs.msdn.com/b/ssdt/>
- Kompatibilitätsübersicht
 - <http://blogs.msdn.com/b/analysisservices/archive/2014/04/03/sql-server-data-tools-business-intelligence-for-visual-studio-2013-ssdt-bi.aspx>
- Übersicht und Einstieg
 - <https://msdn.microsoft.com/de-de/data/gg427686.aspx>

■ Seminare

- Datenbankentwicklung mit SQL Server, SQL Server Reporting Services, SQL Server Integration Services, Migration Access nach SQL Server
 - Termine unter www.berndjungbluth.de

■ Danke

Noch Fragen?

Vielen Dank für die Aufmerksamkeit!