
Data-Tier Application Project in Visual Studio 2010 and SQL Server 2008 R2

Joy Rathnayake (MVP/ MCT)

Mentor - Solid Quality Mentors

<http://www.sharepoint119.com>

<http://www.joyr.ws>

Agenda

- Data-Tier development challenges
- Working with Data-Tier Applications

Data-Tier Dev Challenges

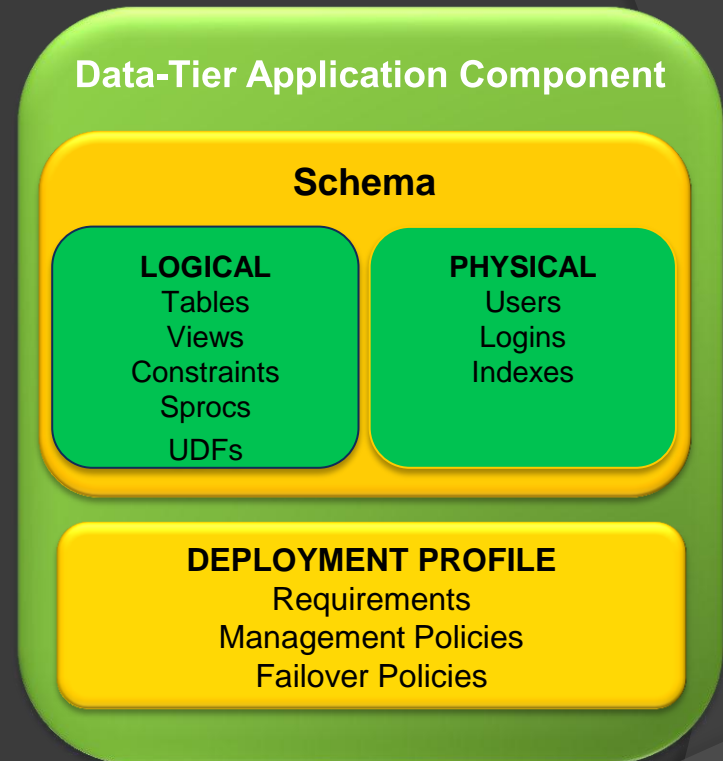
- Limited T-SQL development tooling in earlier Visual Studio versions
- Difficult deployment
 - Need to package applications with databases
 - Need to record deployment intent
 - Handing off T-SQL scripts is error-prone
 - Upgrades are challenging

Data Application Vision

- ◎ T-SQL development experience should be
 - First-class during development
 - Productive
 - Based on understanding of deployment intent
- ◎ Common tooling from Visual Studio 2010
 - Rich T-SQL editor
 - Intellisense
 - Debugger
 - Deep integration with project system

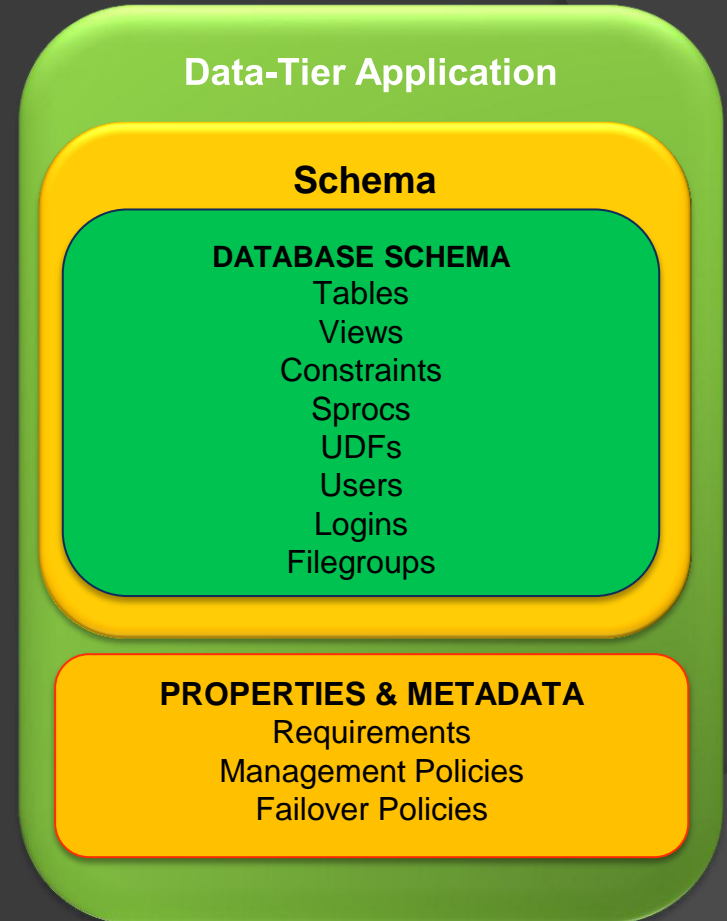
Data-Tier Application Component

- Unit of deployment for T-SQL applications
- Defines contents of the application
- Wraps deployment intent as policies
- Simplifies deployment and administration
 - Install
 - Uninstall
 - Upgrade
 - Repair (not in V1)



Data-Tier Application

- ⦿ Deployed instance of a Data-tier application component
- ⦿ Maps to a database
- ⦿ Has direct and strict relationship with .dacpac file definition



Data-Tier Applications in Visual Studio 2010

- Business goals and vision
 - First-class T-SQL/DAC development experience in VS
 - “Develop, Deploy, Manage”
 - Common tools and experiences in SSMS and VS
 - Editor, IntelliSense, debugging, ...
- Key scenarios
 - Deep integration with project system
 - Highly productive T-SQL editor (IntelliSense, debugger, ...)
 - DAC Projects in Pro+ SKUs
 - Application Lifecycle Management scenarios in Premium & Ultimate SKUs

DAC Projects Vs Database Projects

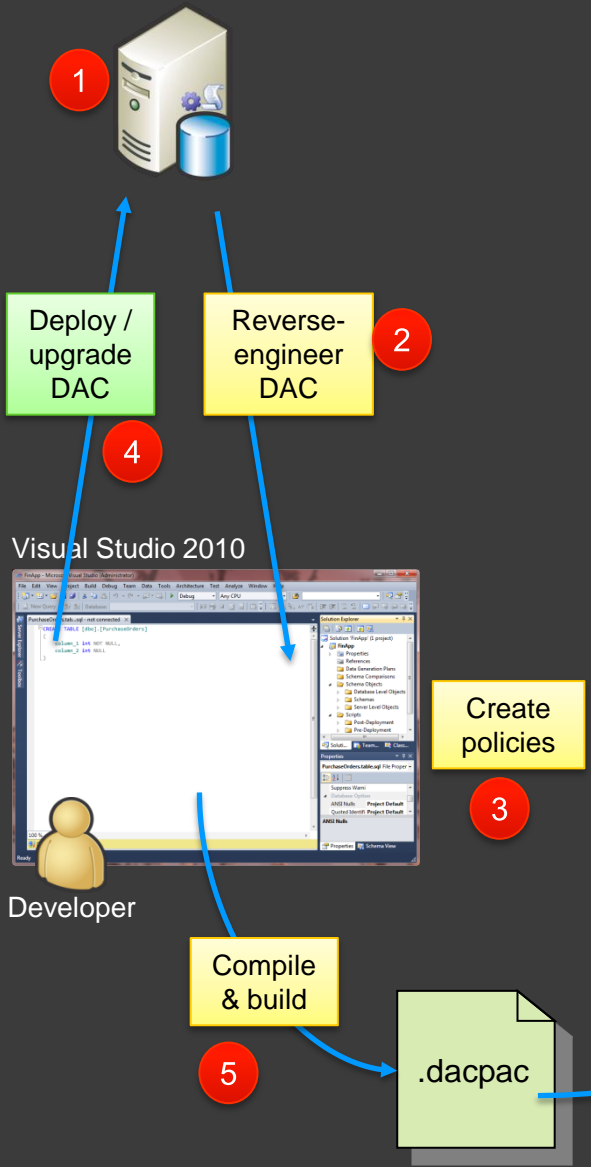
	Database Projects	Data-Tier Application Projects
Target applications	Mission or business critical	Departmental (for V1)
Database support	SQL Server 2005, 2008 & 2008 R2 and third-party databases	SQL Server 2008 R2 (with plans to support SQL Server 2008)
Deployment intent	N/A	Application & deployment intent
Schema complexity	Mission or business critical	Departmental (optimized for up to 1000 objects per project in V1)
Upgrade (schema + data)	Deploy generates SQL scripts; data stays in place	Upgrade is automated; data is migrated
SQL Server object types	All SQL Server 2008 objects	Most common SQL Server objects
T-SQL editing experience	Identical	

Database projects and Data-tier Application projects can operate side by side

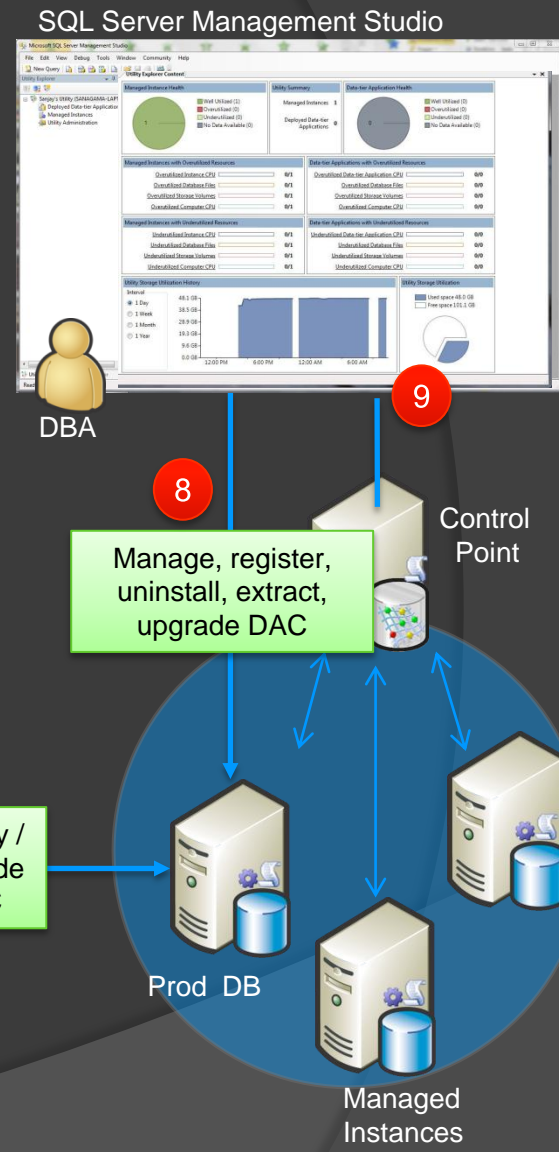
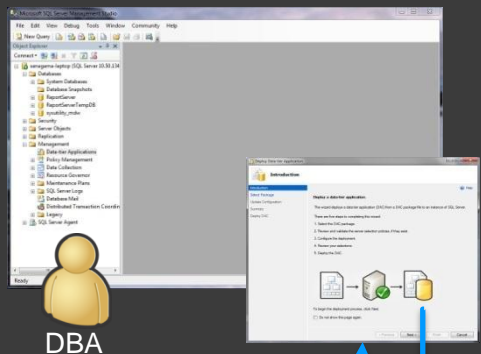
DEVELOP

DEPLOY

MANAGE



SQL Server Management Studio



Supported Objects

Views	Database roles
Clustered indexes	Schemas
Non-clustered indexes	Logins
T-SQL stored procedures	Users
Check constraints	Computed columns
Primary key constraints	Triggers (DML)
Foreign key constraints	Alias types
Unique constraints	Table types
Table-valued functions	Scalar functions

Upgrading Data-Tier Applications

- Upgrade Data-Tier Application Wizard
 - Changes schema and properties of deployed applications
 - New database created with new schema
 - Application name must match currently deployed file
 - SQL authentication logins stored without password and logins are created disabled
 - Windows authentication logins OK as they are
- Data is migrated
 - Original database set to read-only
 - Must consider space usage
 - Both databases then renamed
- Can provide further tailoring via PowerShell

Questions

Thank You..!