

# Creating New Scripts with Test Driven Development

---



**Robert C. Cain, MVP**

OWNER, ARCANE TRAINING AND CONSULTING

@arcanecode [www.arcanecode.com](http://www.arcanecode.com)





## Module 1

- Hortense, PodcastSight, No Agenda Show, Pester Basics

## Module 2

- Describe, Context, It. Module validation.

## Module 3

- Testing functions, Mock Data, Scoping

## Module 4

- More function testing, InModuleScope, Mock Functions

## Module 5

- Changing existing functions



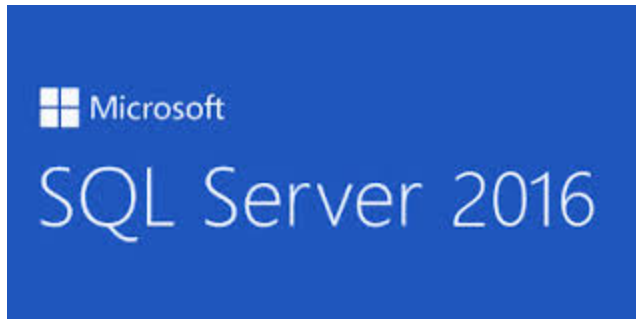


## Podcast-Database Module

- Test-PodcastDatabase
- New-PodcastDatabase
- Test-PodcastTable
- New-PodcastTable
- Update-PodcastTable



Test Driven Development (TDD)  
Domain Specific Language (DSL)



SQL Server 2012, 2014, 2016

Instances of SQL Server

One Default Instance

Same name as the server

PodcastSight will use the default instance

Account running Podcast-Data will have proper permissions to the server

SQL Server Express



Click SELECT ALL  
before clicking Next

SQL Server 2014 Setup

### Feature Selection

Select the Express features to install.

License Terms

Global Rules

Product Updates

Install Setup Files

Install Rules

**Feature Selection**

Feature Rules

Instance Configuration

Server Configuration

Database Engine Configuration

Reporting Services Configuration

Feature Configuration Rules

Installation Progress

Complete

Features:

**Instance Features**

- ☒ Database Engine Services
- ☒ SQL Server Replication
- ☒ Full-Text and Semantic Extractions for Sea
- ☒ Reporting Services - Native

Shared Features

- ☒ Client Tools Connectivity
- ☒ Client Tools Backwards Compatibility
- ☒ Client Tools SDK
- ☒ Documentation Components
- ☒ Management Tools - Basic
- ☒ Management Tools - Complete
- ☒ SQL Client Connectivity SDK

Select All   Unselect All

Instance root directory: C:\Program Files\Microsoft SQL Server\ ...

Shared feature directory: C:\Program Files\Microsoft SQL Server\ ...

Shared feature directory (x86): C:\Program Files (x86)\Microsoft SQL Server\ ...

Feature description:

The configuration and operation of each instance feature of a SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on

Prerequisites for selected features:

Already installed:

- Windows PowerShell 2.0
- Microsoft .NET Framework 3.5

Disk Space Requirements

Drive C: 3815 MB required, 93137 MB available

< Back   Next >   Cancel   Help



SQL Server Express  
installs using a named  
instance.  
  
DON'T TAKE THIS  
OPTION!

The screenshot shows the 'Instance Configuration' window of the SQL Server 2014 Setup. The window title is 'SQL Server 2014 Setup'. The main heading is 'Instance Configuration', with a subtitle: 'Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.'

On the left is a navigation pane with the following items: License Terms, Global Rules, Product Updates, Install Setup Files, Install Rules, Feature Selection, Feature Rules, **Instance Configuration** (selected), Server Configuration, Database Engine Configuration, Reporting Services Configuration, Feature Configuration Rules, Installation Progress, and Complete.

The main area contains the following configuration options:

- ☐ Default instance
- ☒ Named instance:
- Instance ID:
- SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS
- Reporting Services directory: C:\Program Files\Microsoft SQL Server\MSRS12.SQLEXPRESS
- Installed instances:

Below the 'Installed instances' label is a table with the following columns: Instance Name, Instance ID, Features, Edition, and Version. The table is currently empty.

At the bottom right are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.



Change the instance  
at the top to Default  
Instance, then click  
Next

SQL Server 2014 Setup

### Instance Configuration

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

License Terms  
Global Rules  
Product Updates  
Install Setup Files  
Install Rules  
Feature Selection  
Feature Rules  
**Instance Configuration**  
Server Configuration  
Database Engine Configuration  
Reporting Services Configuration  
Feature Configuration Rules  
Installation Progress  
Complete

☒ Default instance  
☐ Named instance: MSSQLSERVER

Instance ID: MSSQLSERVER

SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER  
Reporting Services directory: C:\Program Files\Microsoft SQL Server\MSRS12.MSSQLSERVER

Installed instances:

Instance Name	Instance ID	Features	Edition	Version
---------------	-------------	----------	---------	---------

< Back   Next >   Cancel   Help





# Use Mixed Mode Authentication on the local developer machine

The screenshot shows the 'Database Engine Configuration' window in the SQL Server 2014 Setup. The window has a title bar 'SQL Server 2014 Setup' and a subtitle 'Database Engine Configuration'. Below the subtitle is the instruction: 'Specify Database Engine authentication security mode, administrators and data directories.' On the left is a navigation pane with the following items: License Terms, Global Rules, Product Updates, Install Setup Files, Install Rules, Feature Selection, Feature Rules, Instance Configuration, Server Configuration, **Database Engine Configuration** (selected), Reporting Services Configuration, Feature Configuration Rules, Installation Progress, and Complete. The main area has four tabs: 'Server Configuration' (selected), 'Data Directories', 'User Instances', and 'FILESTREAM'. The 'Server Configuration' tab contains the following content: 'Specify the authentication mode and administrators for the Database Engine.'; 'Authentication Mode' with two radio buttons: 'Windows authentication mode' (unselected) and 'Mixed Mode (SQL Server authentication and Windows authentication)' (selected); 'Specify the password for the SQL Server system administrator (sa) account.' with two password fields labeled 'Enter password:' and 'Confirm password:', both containing masked characters; 'Specify SQL Server administrators' with a list box containing 'ARCANE-VM\arcan\_000 (arcan\_000)' and buttons 'Add Current User', 'Add...', and 'Remove'; and a text box on the right stating 'SQL Server administrators have unrestricted access to the Database Engine.' At the bottom are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

SQL Server 2014 Setup

### Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

License Terms  
Global Rules  
Product Updates  
Install Setup Files  
Install Rules  
Feature Selection  
Feature Rules  
Instance Configuration  
Server Configuration  
**Database Engine Configuration**  
Reporting Services Configuration  
Feature Configuration Rules  
Installation Progress  
Complete

Server Configuration | Data Directories | User Instances | FILESTREAM

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

☐ Windows authentication mode

☒ Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password: .....

Confirm password: .....

Specify SQL Server administrators

ARCANE-VM\arcan\_000 (arcan\_000)

Add Current User | Add... | Remove

SQL Server administrators have unrestricted access to the Database Engine.

< Back | Next > | Cancel | Help



# Requirements – Module



**Podcast-Database.psm1**

**Manifest file (psd1) that references the psm1**

**Valid PowerShell Code**

**Functions should exist in same folder**

# Requirements - All Functions



## Help

- Synopsis, Description, Example

## Write-Verbose

## Advanced Functions

- param block, pipeline enabled

## Functions in their own files

## Valid PowerShell

# Requirements – Test-PodcastDatabase



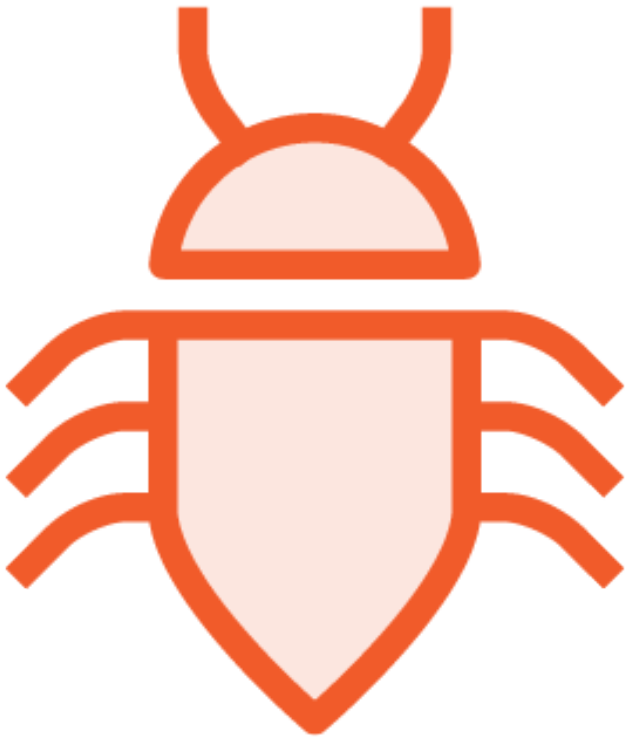
**Validate the existence of a database**

**Parameter PodcastDatabaseName**

**Default value of PodcastSight**

**Return \$true if it exists, else \$false**

# Tests – Test-PodcastDatabase



**Validate returns \$false for database name that is not present**

**Validate returns \$true when an existing database is passed as a parameter**

**Validate returns \$true when using the default database**

# Testing Database Scripts



**Mocks not really effective**

**Option 1: Connection string changes based on acceptance vs unit test**

**Option 2: Connection string points to 'localhost', the computer where the tests are being run**

**Hortense uses option 2 for simplicity**

# Requirements – New-PodcastDatabase



**Create a new database**

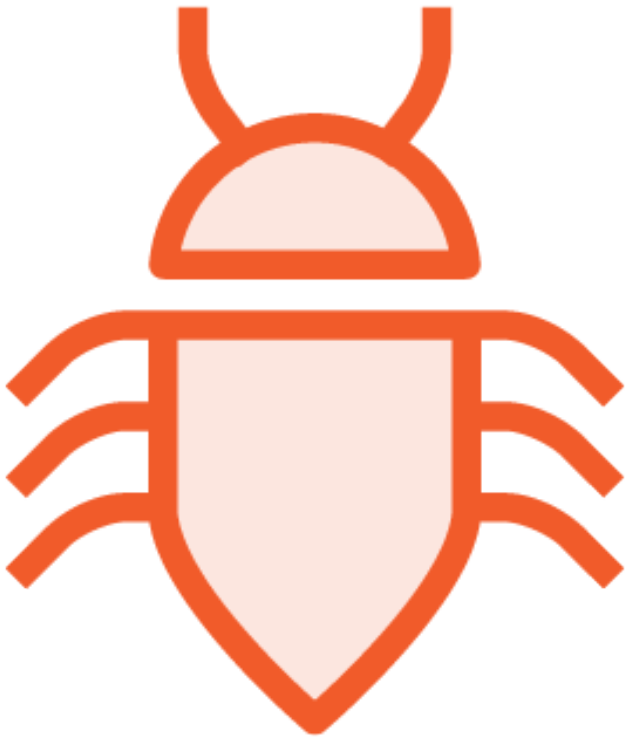
**Parameter: DatabaseName**

**Create one staging table in the database**

- Title, ShowUrl, EmbeddedHTML, Hosts, PublicationDate, ImageUrl, AudioUrl, AudioLength

**Have a stored procedure InsertStagingData**

# Tests – New-PodcastDatabase



## **For PodcastSight (the default):**

- Create it, or verify it exists
- Verify staging table exists
- Validate column names
- Verify Stored Proc exists

## **For a database that does not exist**

- Same as above, but database must not exist beforehand

## **Drop the test database**



# Requirements – Test-PodcastTable



## **Validate the existence of a table**

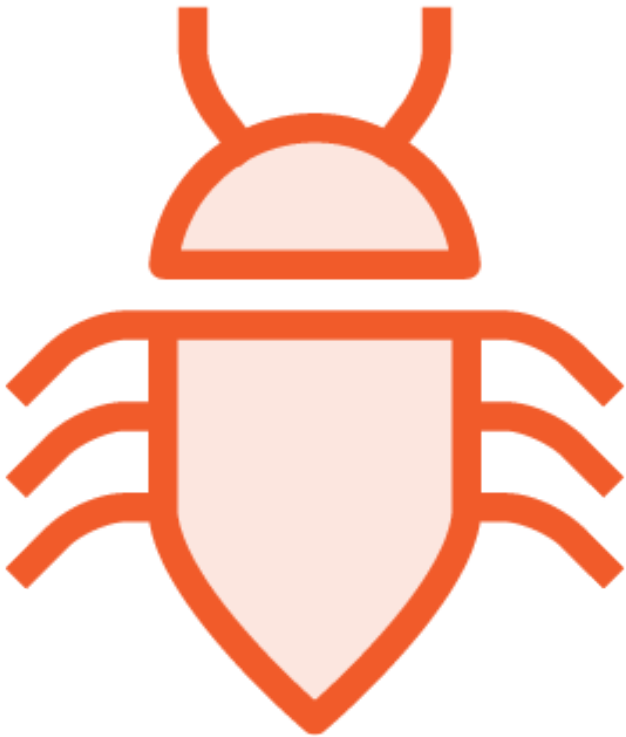
### **Two parameters**

- Name of DB to look in, default to PodcastSight
- Name of Table to look for

**Use default schema of dbo if no schema included in table name**

**Return \$true if it exists, else \$false**

# Tests – Test-PodcastTable



## Validate:

### Existence of PodcastSight

Staging exists using default db name

dbo.Staging exists using default db name

Staging exists passing in db name

dbo.Staging exists passing in db name

Returns false when a nonexistent table name is passed in

# Requirements – New-PodcastTable



**Validate table does not exist before creating**

**Create a new table with the columns:**

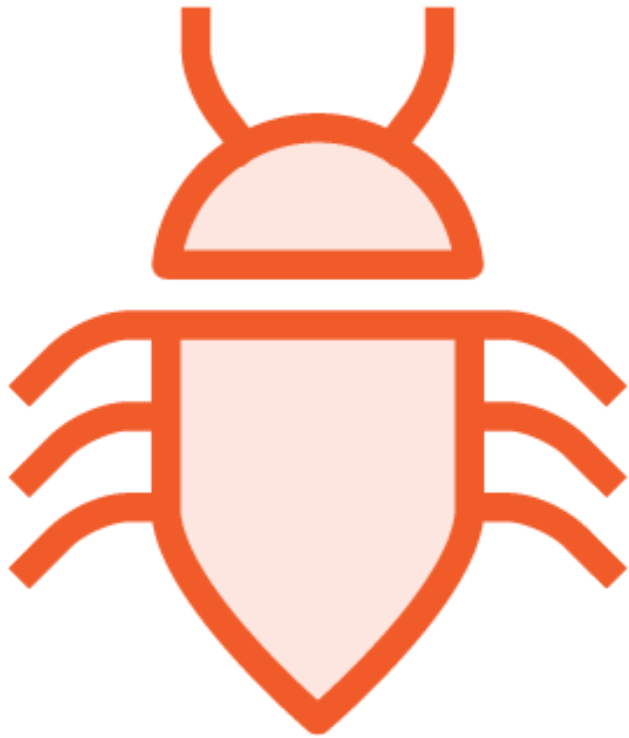
- Title, ShowUrl, EmbeddedHTML, Hosts, PublicationDate, ImageUrl, AudioUrl, AudioLength

**Two parameters**

- DB to use, default to PodcastSight
- Name of Table to create

**Use default schema of dbo if no schema included in table name**

# Tests – New-PodcastTable



**Mock Test-PodcastTable**

**Validate a table is created**

**Validate the table has the correct fields**

# Requirements – Update-PodcastTable



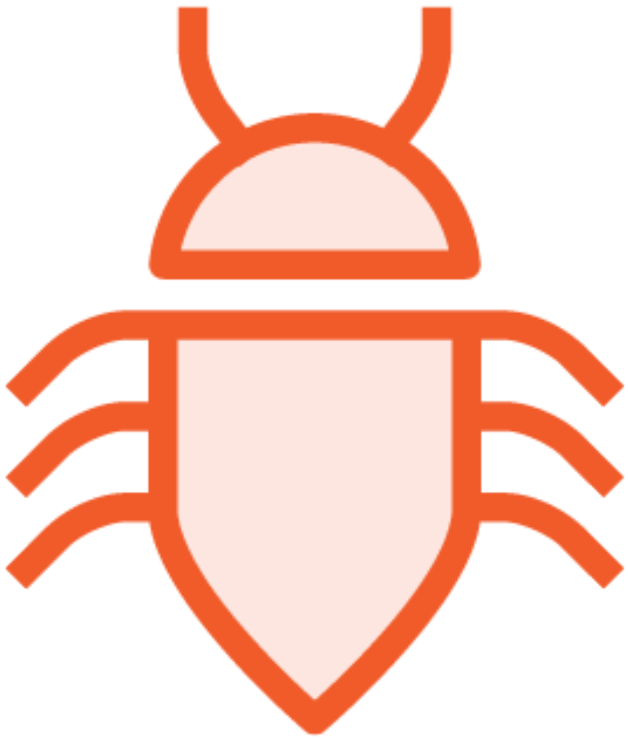
## Three parameters:

- DB Name (default to PodcastSight)
- Table Name. Use dbo for default schema
- Podcast Data (from Get-PodcastData)

**Update Staging Table using the InsertStagingData stored procedure, after ensuring the staging table exists**

**User T-SQL Merge to update target table from Staging**

# Tests – Update-PodcastTable



## Unit Tests

- Mocked RSS Data
- Update newly created test table
- Compare each column for each row in the source to what is placed in the db

## Acceptance Tests

- Real RSS Data
- Existing table for the podcast
- Compare each column for each row in the source to what is placed in the db

# Summary



**Created a new module using TDD**

**Database application testing issues**

**Helper functions for tests**

# Summary



**Created comprehensive tests for existing code**

**Pester as a guide for change requests**

**Pester + TDD to create a new module**