Schema Management

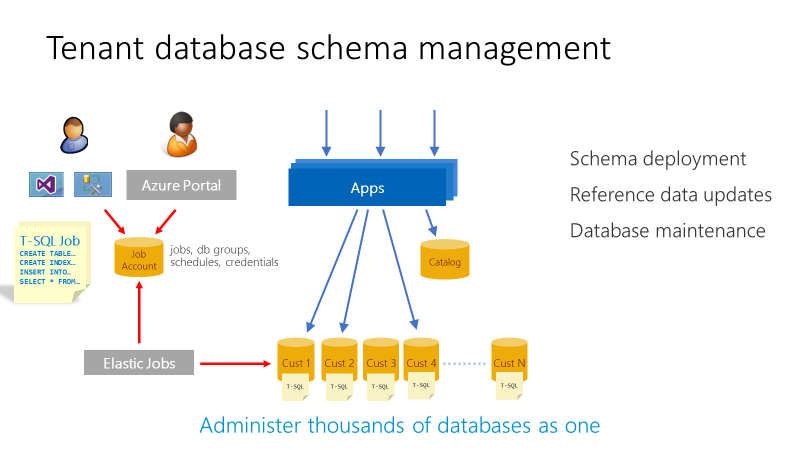
A ‘SaaS in a Box’ Tutorial

# Purpose of this tutorial

The Introduction to the WTP Application tutorial shows how the WTP app can provision a tenant database with its initial schema and register it in the catalog, making it available to the rest of the application. Like any application, the WTP app will evolve over time, and will at times require changes to the database. This may include new or changed schema, new or changed reference data, as well as routine database maintenance tasks to ensure optimal app performance. With a SaaS application, these changes need to be deployed in a coordinated manner across a potentially massive fleet of tenant databases and must be incorporated into the provisioning process for new tenant databases. This tutorial explores two scenarios, deploying reference data updates which must be shared by all tenants, and retuning an index on the table containing the reference data. The Elastic jobs feature is used to execute these operations across all the tenants and a ‘golden’ tenant database that is used as a template for new databases.

# Introduction to the SaaS Schema Management patterns

The single tenant per database SaaS patterns benefits in many ways from the data isolation that results, but at the same time introduces additional management complexity stemming from the need to maintain and manage many databases. Elastic Jobs facilitates administration and management of the SQL data tier. Jobs enable database administrators and/or operations to securely and reliably, run tasks (T-SQL scripts) independent of user interaction or input, against a group of databases. This method can be used to deploy schema and common reference data changes across all tenants in the application. In addition, it can be used to maintain a ‘golden’ copy of the database used to create new tenants, ensuring it always has the latest schema and reference data.



# Elastic Jobs private preview

The ‘new’ Elastic Jobs is integrated as a feature in Azure SQL Database and requires no additional Azure services, compared to the Public Preview customer-hosted version of Elastic jobs. In scope for the Private Preview is PowerShell to create the job account, and T-SQL to create and manage jobs. Public Preview will broaden capabilities to Portal, PowerShell and REST APIs in addition to new features.

In this tutorial, a new job account database and job account will be created. The tutorial then walks you through creating jobs to deploy new reference data and manage indexes.

# Prerequisites

1. Subscription must be registered to use the Elastic jobs Private Preview. Send email for [SaaSFeedback@microsoft.com](mailto:SaaSFeedback@microsoft.com) to request registration. Once your subscription is confirmed registered, you can also verify registration by calling the Get-AzureRmProviderFeature cmdlet with the following parameters:
   * Get-AzureRmProviderFeature -ProviderName Microsoft.Sql -FeatureName sqldb-JobAccounts
2. To update your Azure PowerShell cmdlets to include the new Elastic jobs APIs, you must download and install the ‘updated’ latest AzurePowerShell.msi located here: <https://github.com/jaredmoo/azure-powershell/releases>

Setup

Download and extract **WTPLearningModules.zip** to a convenient folder.

Deploy the **WTP Application.** Ensurethe catalog is initialized using the Demo Assistant app. See the Introduction to the WTP SaaS Application tutorial for deployment instructions.

**SSMS** can be used to explore database schema and execute SQL queries directly. In this tutorial, SSMS will be used to create and manage jobs.

**PowerShell ISE** is recommended to execute scripts and follow their execution in debug mode.

**PowerShell Tips**

* Open and configure demo- scripts in the PowerShell ISE.
* Use F5 to run the script (using F8 is not advised as the $PSScriptRoot variable is not evaluated when running snippets of a script).
* Use F9 to set a breakpoint to let you trace the script in debug mode to see how it works
* Use F10 to step through the script, F11 to step into a function, and Shift-F11 to step out.

# Walkthrough

## Getting Started – create a new job account database and new job account

This tutorial requires you use PowerShell to create the job account database and job account. Like MSDB and SQL Agent, Elastic Jobs uses an Azure SQL Database to store job definitions and job status and history. Once, the job account is created, you can start to create and monitor jobs immediately.

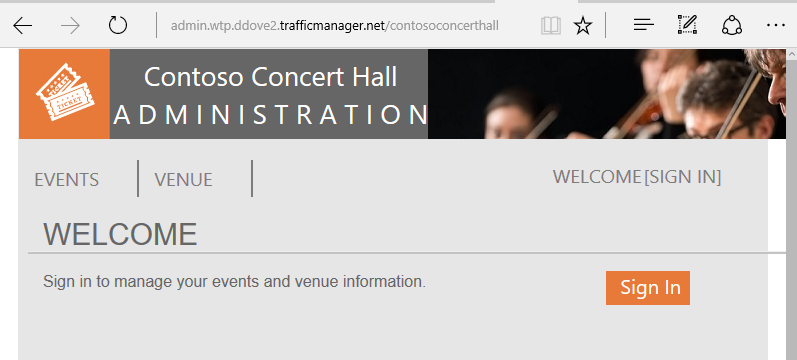
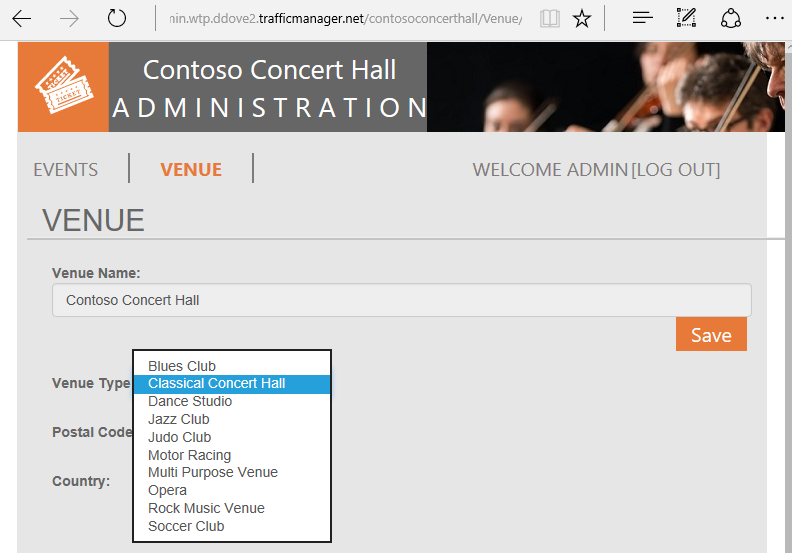
1. Update the user configuration file used by all tutorial scripts. Update again if you redeploy the app.
   1. Open ...\Learning Modules\UserConfig.psm1 in **PowerShell ISE**
   2. Modify **$userConfig.ResourceGroupName** to the resource group used for the deployed app.
   3. Modify **$userConfig.Name** to the User name used for the deployed app.
2. Open …\Learning Modules\Schema Management\**Demo-SchemaManagement.ps1** in PowerShell ISE.
3. Execute the script using **F5**.

The Demo-SchemaManagement script calls the Deploy-SchemaManagement script to create an S2 database named **jobaccount** on the catalog server. It then creates the job account, passing the jobaccount database as a parameter to the job account creation call.

## Exercise 1: Create a job to deploy new reference data to all tenants

The WTP database includes a set of venue types that define the kind of events that are hosted at a venue. The set of supported venue types is held in the VenueTypes table which is present in every tenant database. These venue types correspond to the background image you see in the WTP events and admin apps. In this exercise, you will deploy an update to all the tenant databases to add two additional venue types, ‘Motorcycle Racing’ and ‘Swimming Club’.

When prior provisioning step completed it opened the WTP admin page in the browser for the tenant ‘Contoso Concert Hall’. Click the **Sign in** button, and again click the **Sign in** button to sign in to the admin site using the default password. Click the Venue Type drop down menu and validate that only 10 venue type options are available, and specifically that ‘Motorcycle Racing’ and ‘Swimming Club’ are not included in the list.

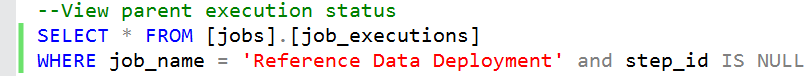
Now let’s create a job to update venue types table in all the tenant database and add ‘Motorcycle Racing’ and ‘Swimming Club’.

To create a new job, we use a set of jobs ‘system’ stored procedures created in the jobaccount database when the job account was created.

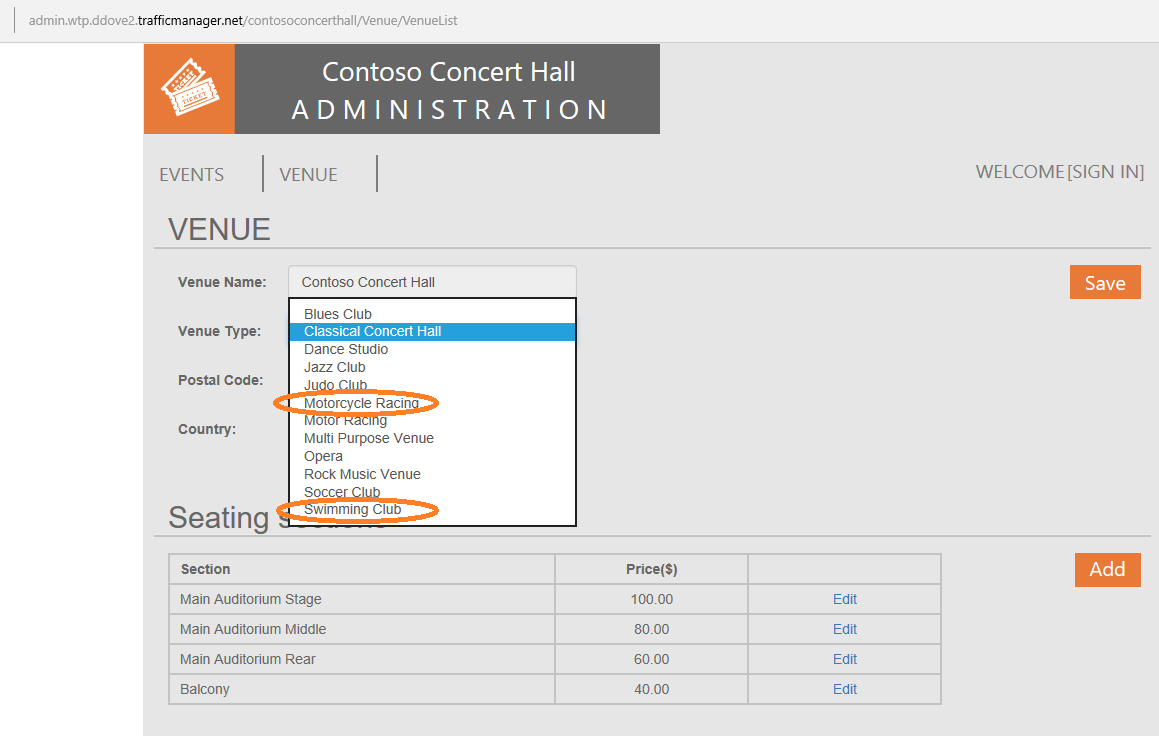
1. Open SSMS and connect to the catalog-<WtpUser>.database.windows.net server
2. Open the file …\Learning Modules\Schema Management\DeployReferenceData.sql
3. Modify <WtpUser>, use the user name used when you deployed the WTP app
4. Right click, select **Connection**, and connect to the catalog-<WtpUser>.database.windows.net server, if not already connected
5. Ensure you are connected to the jobaccount database and press **F5** to run the script
   * **sp\_add\_target\_group** creates the target group name DemoServerGroup, now we need to add target members.

* **sp\_add\_target\_group\_member** adds a *server* target member type which deems all databases within that server (note this is the customer1-<WtpUser> server containing the tenant databases) at time of job execution should be included in the job, the second is adding a *database* target member type, specifically the ‘golden’ database, baseTenantDB which resides on catalog-<WtpUser> server, and lastly another *database* target group member type to include the adhocanalytics database that will be used in a later tutorial.
* **sp\_add\_job** creates a new job called “Reference Data Deployment”
* **sp\_add\_jobstep** creates the job step containing T-SQL command text to update to the reference table, VenueTypes
* The remaining views in the script display the existence of the objects and monitor job execution.

1. Re-run the query view parent execution status until the job has completed
   * select \* from [jobs].[job\_executions] where job\_name = 'Reference Data Deployment' and step\_id IS NULL
   * Review the status value from the lifecycle column to monitor the status. Once, Succeeded, the job has successfully finished on all tenant databases and the two additional databases containing the reference table.



Go back to the browser with the admin site opened, refresh the browser and select the Venue Types drop-down list again to see the two new types added to the tenant database.



## Exercise 2: Create a job to manage the reference table index

This exercise creates a job to rebuild the index on the reference table primary key, a typical database management operation an administrator might perform after a large data load into a table.

Create a new job using the same jobs ‘system’ stored procedures.

1. Open SSMS and connect to the catalog-<WtpUser>.database.windows.net server
2. Open the file …\Learning Modules\Schema Management\OnlineReindex.sql
3. Right click, select Connection, and connect to the catalog-<WtpUser>.database.windows.net server, if not already connected
4. Ensure you are connected to the jobaccount database and press **F5** to run the script

* **sp\_add\_job** creates a new job called “Online Reindex PK\_\_VenueTyp\_\_265E44FD7FD4C885”
* **sp\_add\_jobstep** creates the job step containing T-SQL command text to update the index

# Additional resources

* To learn more about Elastic Jobs, see …\Learning Modules\Schema Management\Elastic Jobs private preview (User Guide).docx