Unicorn with Sitecore

What is Serialization?

Serialization allows you to serialize an entire Sitecore database or a series of items in a database to text files. You can then use these text files to transfer this database or series of items to another database or Sitecore solution.
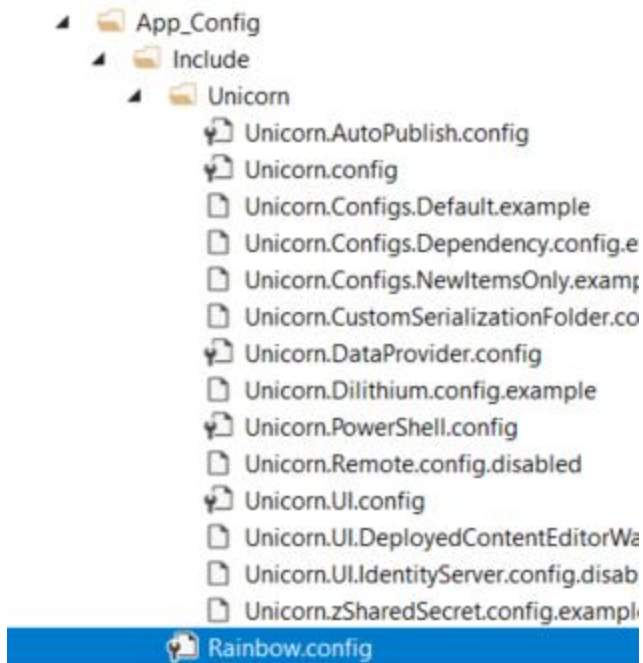
Why to use Serialization?

- The Sitecore serialization functionality is designed to help teams of developers that work on the same Sitecore solution to synchronize database changes between their individual development environments but is also valuable when a single developer works on a solution.
- Sitecore serialization is typically used in combination with a source control system to make it easy for developers to synchronize database changes between their local databases that they use for development — with the added benefit that the source control system will keep track of database changes and allow you to compare database changes.

How to implement Serialization using Unicorn?

Steps:

1. Open Visual Studio.
2. Go to NuGet Update manager
3. Install Latest Version of Unicorn.
4. Then you will see Unicorn related configs are generated under the AppConfig/Include/Unicorn.

```
App_Config
  Include
    Unicorn
      Unicorn.AutoPublish.config
      Unicorn.config
      Unicorn.Configs.Default.example
      Unicorn.Configs.Dependency.config.e
      Unicorn.Configs.NewItemsOnly.examp
      Unicorn.CustomSerializationFolder.co
      Unicorn.DataProvider.config
      Unicorn.Dilithium.config.example
      Unicorn.PowerShell.config
      Unicorn.Remote.config.disabled
      Unicorn.UI.config
      Unicorn.UI.DeployedContentEditorWa
      Unicorn.UI.IdentityServer.config.disab
      Unicorn.zSharedSecret.config.exampl
  Rainbow.config
```

5. Do can edit these config give the project path and target database or create a patch config.

6. Patch file code:

```xml
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
  <sitecore>
    <unicorn>
      <configurations>
        <configuration name="SitecoreUnicorn" description="SitecoreUnicorn description">
          <targetDataStore
physicalRootPath="C:\inetpub\wwwroot\SiteUnicorn\Website\Data\Serialization"
useDataCache="false" type="Rainbow.Storage.SerializationFileSystemDataStore, Rainbow"
singleInstance="true"/>
          <predicate type="Unicorn.Predicates.SerializationPresetPredicate, Unicorn"
singleInstance="true" >

            <!--Tempaltes-->
            <include name="SitecoreUnicorn.Template.Foundation" database="master"
path="/sitecore/templates/Foundation">
                <!-- <exclude children="true"/> -->
            </include>
            <include name="SitecoreUnicorn.Template.Project" database="master"
path="/sitecore/templates/Project">
                <!-- <exclude children="true"/> -->
            </include>
            <include name="SitecoreUnicorn.Template.Feature" database="master"
path="/sitecore/templates/Feature">
                <!-- <exclude children="true"/> -->
            </include>
            <include name="SitecoreUnicorn.Template.Common" database="master"
path="/sitecore/templates/Common">
                <exclude children="true"/>
            </include>
```
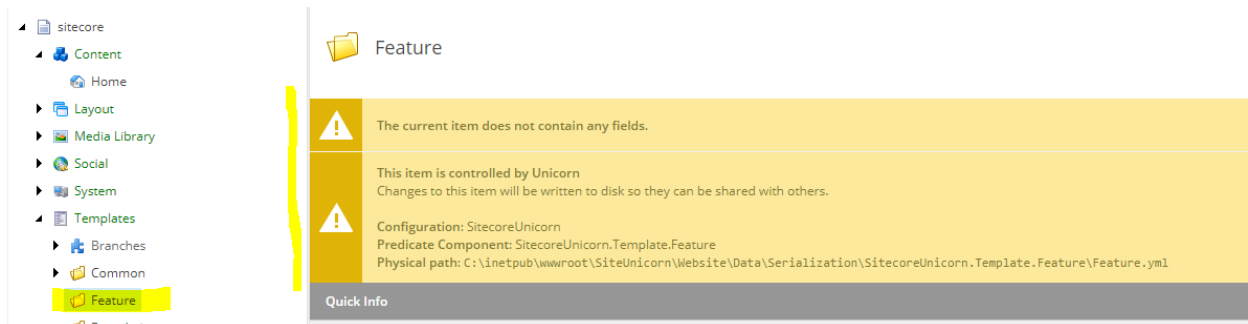
```
        </predicate>
      </configuration>
    </configurations>
  </unicorn>
</sitecore>
  </configuration>
```

7. Change the project physical root path in this config.
8. If you want to add their children also then uncomment the line "`<exclude children="true"/>`"
9. Build and Publish the Solution.
10. Open the Sitecore and select the item which we included and give the item path.

11. You can see the Unicorn change on the item and Physical file generated on the given path.

12. After that open the URL: <DomainName>/unicorn.aspx.

13. Then you need to Sync the item by clicking the Sync button.
14. If you included anything new in the predicate, then firstly you need to Reserialize it first then Sync.