**IGHS – Build PipeLine**

**IGHS Build Pipeline**

**User Guide**

**IGHS Jaguars Team**

|  |  |
| --- | --- |
| Author: | Sathish Ramaraj |
| Contributors: | |
| Project Manager: | |
| Document Version: | 1.0 |
| Initially Created: | 12 Dec, 2014 |
| Last Updated: | 12 Dec, 2014 |
| Signed Off by: | |
| Signed Off Date: | |

[**Deployment Pipeline**](http://geekswithblogs.net/TarunArora/archive/2011/06/06/unshelve-shelveset-created-from-one-branch-to-another.aspx)

The deployment pipeline is the key pattern that enables continuous delivery. A deployment pipeline implementation provides visibility into the production readiness of your applications by giving feedback on every change to your system. It also enables team members to self-service deployments into their environments.

[**Pipeline Overview**](http://geekswithblogs.net/TarunArora/archive/2011/06/06/unshelve-shelveset-created-from-one-branch-to-another.aspx)

The component manifest will be generated for each primary components build/tfs check-in. The manifest generation and Build pipeline are integrated. Jenkin jobs will be triggered based on the schedule. The deployment would be picked during schedule run.

The version of shared libraries which is used in code referencing should also be used to generate the manifest. This is very important as the same version should be used for code reference and deployment. The manifest generation build have flexibility to update the manifest template automatically for shared libraries version from Nuget package config file. The component manifest build initiates the nuget update activity before generating the manifest.

TFS Check-in

Component

Manifest

Deployment

using Nolio CLI

**Manifest Template update from Nuget Package.cofig**

**Initiate Schedule deployment from Jenkin**

**Notification from Jenkin for Deployment status**

**What is primary Build?**

Primary build is one of the MSI package for the component manifest. The primary build will be initiating the manifest generation and the same manifest version should be picked up by build pipeline.

The name of the Primary build definition is given below:

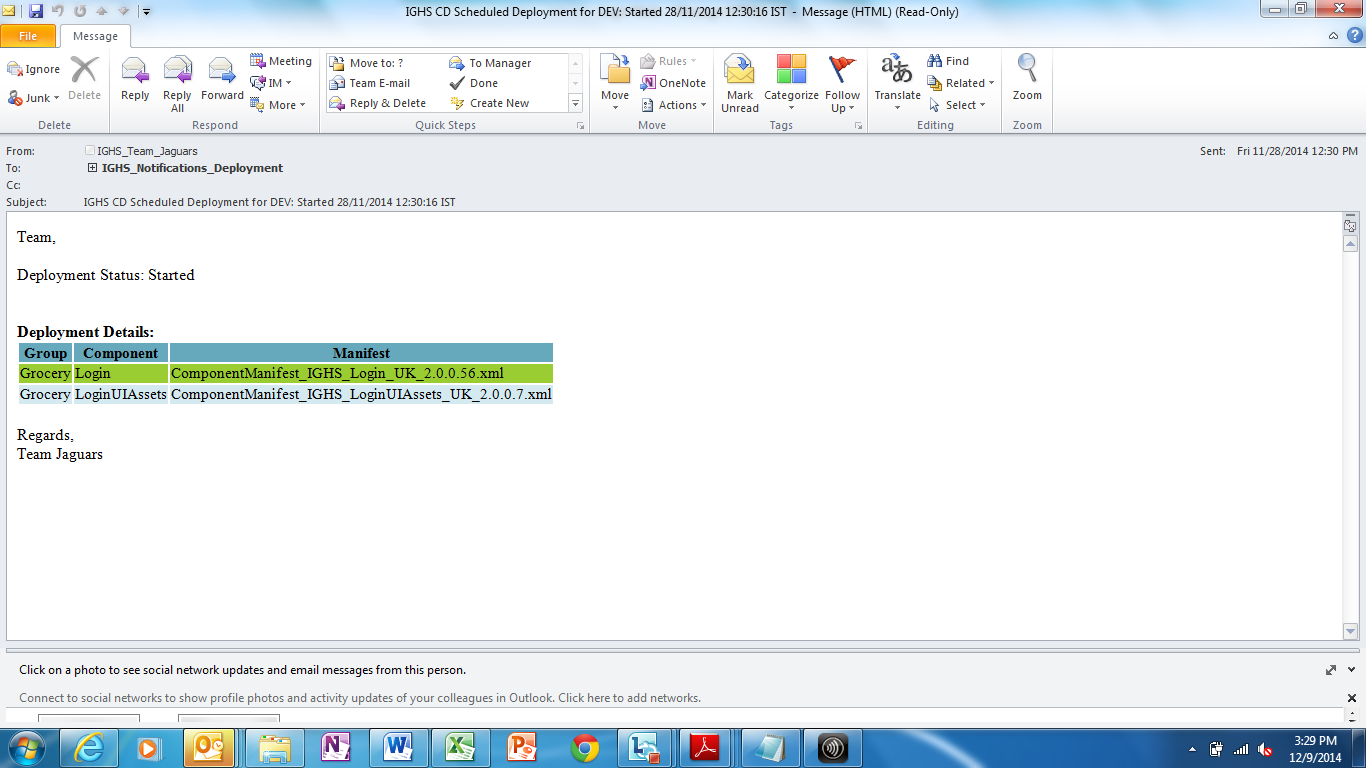
|  |  |  |
| --- | --- | --- |
| **Component Name** | **Primary Build definitions Name** | **Manifest Generation Build Definitions Name** |
| BOA | Tesco.Com.Web.BackOffice | RTC3.0\_Component\_Manifest\_BackOffice\_CD |
| Login | Tesco.Com.Web.Login | RTC3.0\_Component\_Manifest\_Login\_CD |
| LoginUIAssest | Tesco.Com.Web.Login.UIAssets | RTC3.0\_Component\_Manifest\_LoginUIAssets\_CD |
| Checkout | Tesco.Com.Web.UI.Grocery.Checkout | RTC3.0\_Component\_Manifest\_Grocery\_OrderCheckout\_CD |
| CustomerProfile | Tesco.Com.Web.UI.Grocery.MyAccount | RTC3.0\_Component\_Manifest\_Grocery\_CustomerProfile\_CD |
| BasketBuilding | Tesco.Com.Web.UI.Grocery.ShoppingCart | RTC3.0\_Component\_Manifest\_Grocery\_BasketBuilding\_CD |
| Home | Tesco.Com.Web.UI.Grocery.Home | RTC3.0\_Component\_Manifest\_Grocery\_Home\_CD |
| Delivery | Tesco.Com.Web.UI.Grocery.Delivery | RTC3.0\_Component\_Manifest\_Grocery\_Delivery\_CD |
| FindProducts | Tesco.Com.Web.UI.Grocery.Products | RTC3.0\_Component\_Manifest\_Grocery\_FindProducts\_CD |
| GroceryHost | Tesco.Com.Web.UI.GroceryHost | RTC3.0\_Component\_Manifest\_Grocery\_GroceryHost\_CD |

🛈 **If you want to deploy some components without check-in of primary build, you should trigger manually to the manifest generation build. And the new manifest version should get the latest artifacts which will be deployed automatically using build pipeline.**

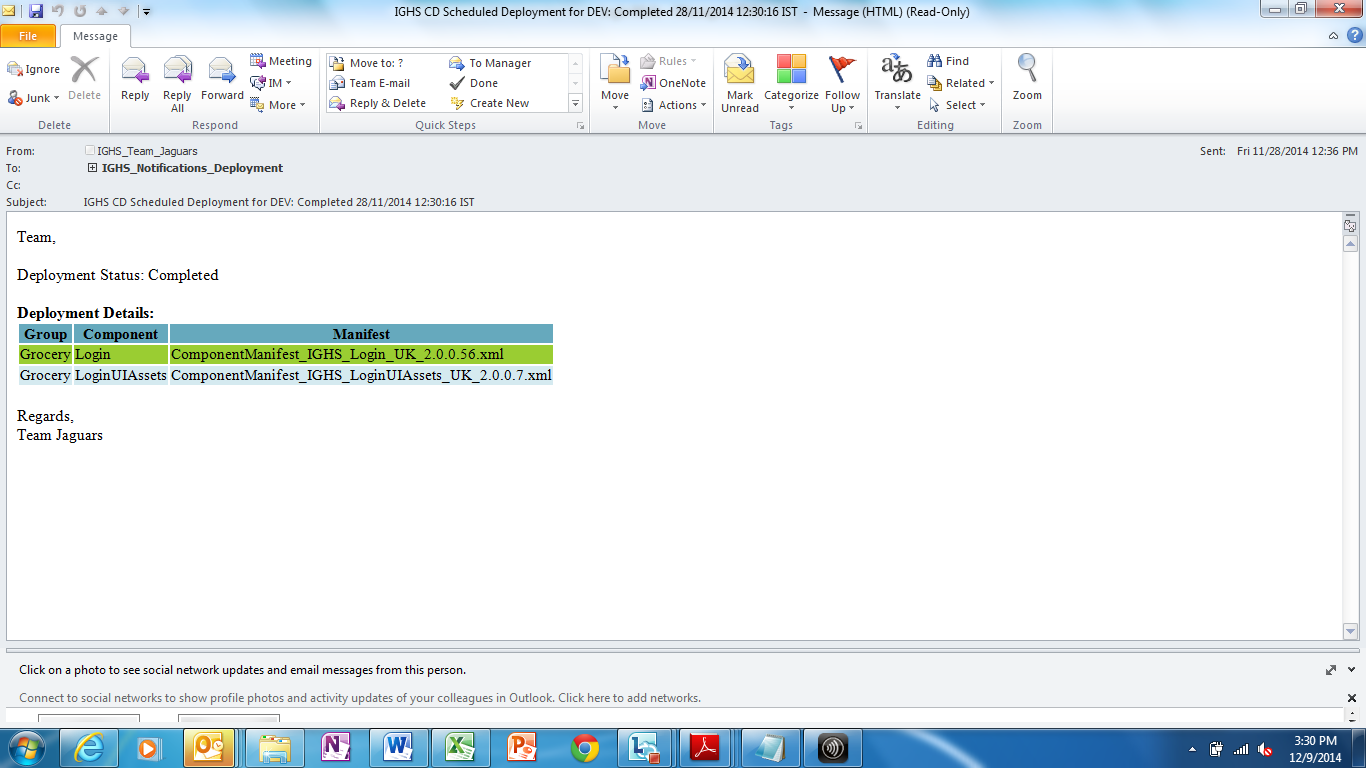
**Dev sheet update**

Every component deployment you will get the nonfiction mail from build pipeline, which is contains deployment status and manifest versions. This manifest version need to be update in the Dev sheet *($/InternationalDeployment/Deployments/R2.0/DEV/ DEV\_Component\_Versions.xls)* manually and would be used for the STG deployment.

***Deployment Started***



***Deployment Started***



**Schedule Deployment for STG**

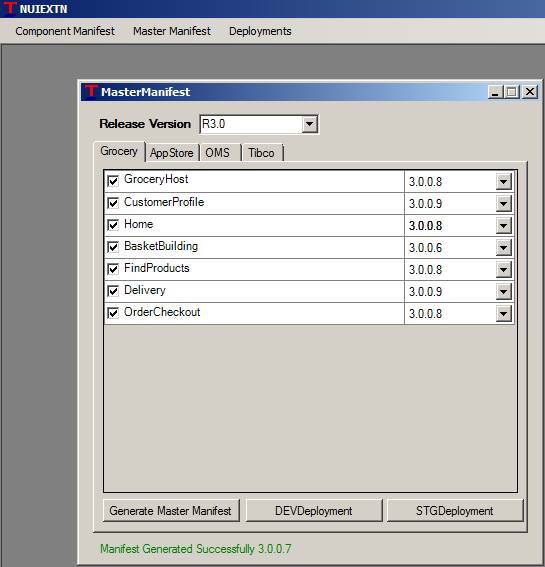
The Scheduled Deployments has been implemented for STG environment. This Deployment would be picked based on the schedule from Nolio tool. The Workflow will get the Manifest from the build output location [*\\Uktee01-clusdb\IGHSBuildOutput\IGHS\_Manifest\ScheduldedDeployment\STG\<ComponentName*](file:///\\Uktee01-clusdb\IGHSBuildOutput\IGHS_Manifest\ScheduldedDeployment\STG\%3cComponentName)*>.*

**Creating Master Manifest for STG Schedule**

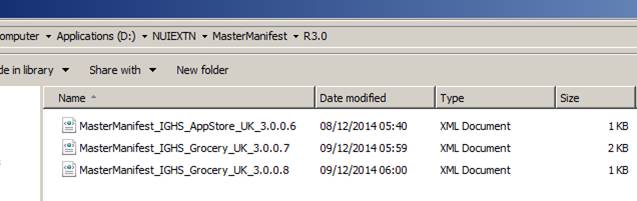
Master manifest generation is required only Grocery/Appstore components. The remaining components will be picked up by the same file.

***Master Manifest Generation:***

**Step 1:** Generate the master manifest from NUIEXTN as given below. If you want to deploy single Grocery/Appstore component, you should mandatory to generate master manifest.



**Step 2:** Generated master manifest will be dropped to your local system *@ D:\NUIEXTN\MasterManifest.*



**Step 3:** These manifest files need to check-in to TFS as given below path details. There is a TFS build configured which have flexibility to copy manifest file to Drop location[*\\Uktee01-clusdb\IGHSBuildOutput\IGHS\_Manifest\ScheduldedDeployment\STG\ <ComponentName*](file:///\\Uktee01-clusdb\IGHSBuildOutput\IGHS_Manifest\ScheduldedDeployment\STG\%20%3cComponentName)*>.* And manifest files would be used for the STG deployment.

|  |  |
| --- | --- |
| **Component Name** | **TFS Path for Scheduled Deplyment Manifest** |
| Appsotore | $/InternationalDeployment/ScheduldedDeployment/STG/AppStore |
| BOA | $/InternationalDeployment/ScheduldedDeployment/STG/BOA |
| Grocery | $/InternationalDeployment/ScheduldedDeployment/STG/Grocery |
| Login | $/InternationalDeployment/ScheduldedDeployment/STG/Login |
| LoginUIAssets | $/InternationalDeployment/ScheduldedDeployment/STG/LoginUIAssets |
| UIAssets | $/InternationalDeployment/ScheduldedDeployment/STG/UIAssets |

**Note:** If you want to deploy Other Components (BOA,Login,LoginUIAssets) which is not mandatory to generate Master Manifest. These component manifest file will able available in Dev environment, you should check-in to TFS for the same if you want STG schedule.

**!**