**Standard Operating Procedure (SOP)**

|  |  |
| --- | --- |
| **Atlas Feature Releases** | |
| **Context** | This is the SOP is to deploy Atlas Features to the Production environment using Release Management (RM). **This SOP does not cover Atlas Container deployments.** |
| **Purpose** | To list steps that Atlas Feature Development Team members must follow to deploy Features via RM to Production. Note that a subsequent database script needs to be run by Data Services to enable the Feature in Production. |
| **High level process overview** | * Approve Release(s) to Production using RM Client * Monitor Release and confirm successful execution of the RM Release Template * Merge source control branches as appropriate. * (Data Services) Run database script to add/enable the new/updated Feature in Production. |
| **Roles and responsibilities** | This SOP is to be performed by a member of the Feature Development Team. The **Atlas Features – Feature Developers** RM Role is required to Approve Releases from QA to Production. |
| **Business rules** | * Once a feature has been published to production it may not be updated with another deployment – a new version of the feature must be deployed. * Deploying a Feature to the production environment does not mean that it is immediately available to users - it first needs to be enabled through platform services. Therefore Feature deployment is a two-step process: (i) Feature Deployment, and (ii) Feature Configuration. The Feature is only Live after both steps have been performed. * The Feature Deployment step (this SOP) can be done at any time. * The Feature Configuration step can only be done once the Feature Deployment step has been done, as this step will make the Feature Live. * A Feature comprises 2 main components: (i) html, and (ii) feature-specific JavaScript. * A Feature also has 2 main dependencies: (i) common content – referred to as a “Feature Application”, and (ii) Themes – the look and feel of a Feature. Feature Applications and Themes are deployed independently and would ideally be deployed before the dependent Features are deployed. |
| **Procedures** | Refer to steps below. |

# Feature Deployment via Release Management

## Prerequisites

* The ID of the Release to deploy must be on hand
* The associated Release Configuration data script must be committed to the $DataServices repository. This script must already have been run in QA.
* Signed-off Release Item covering this Release Deployment and the associated Release Configuration.

## Steps

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Launch the Release Management Client desktop application. |
| 2 | Navigate to the Releases screen: |
| 3 | Select the appropriate Release (identify via the Release Id) and confirm that the Release has a Current Stage of **Production** and a Current Step of **Accept Deployment**. |
| 4 | Press the Open button (or double-click the Release) to open the Release. If there are pending approvals for the Release you will be presented with the **My Approval Requests** view for the Release. |
| 5 | Select the Release item and press the **Approve** button to promote the Release to Production. The Release screen will then switch to the (default) View Log view. |
| 6 | Monitor the deployment progress from the Release View Log screen. On completion the status of the Release will change to **Production – Validate Deployment**. |
| 7 | Confirm successful deployment of the Feature.  *On way to do this is by manually navigating to the html page of the new Feature version you have deployed. The Feature wont load, per se, but you can confirm at least that the Feature content was successfully deployed.* |
| 8 | Validate deployment of the Release by opening the release (as per step 4), selecting the Release item (as per step 5) and pressing the **Approve** button. This will then set the status of the Release to **Released**. |
| 9 | Synchronise the Source Code repository for the Feature by merging the Development branch to the Master branch. |
| 10 | Coordinate the execution of the Feature Configuration script with Data Services. |

Procedure Information

|  |  |
| --- | --- |
| **Procedure Owner** | SA Features Team |
| **Procedure Author** | Daniel Morris |

Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | 1 | **Version Date** | 25/06/2015 |