Content Management System: Content Workflow

# Overview

The content management system supports versioned content which it manages and promotes between multiple environments by a workflow involving approvals and status of content elements. Because the content lifecycle is managed by this workflow, segmentation of content into isolated databases is unnecessary and content operations will be performed on the production instance. This enables greater visibility and higher cohesion for management of content production and promotion, as well as reducing complexity for the promotion of content through environments.

# Content Environments

As content approval and status is modified, the CMS will consider it eligible for display in a given environment. Content environments are ranked in a hierarchy which is used to help select which content is visible to which environment, using a probing rule for selection. Content selection preference is given to the version that has the closes rank to that environment, and if that is not found then selecting the content for the next closest environment of higher rank. The following content environments, from highest to lowest rank, are recognized:

* **Production**—intended to model the consumer application’s production environment. Only content that is recognized as production is visible in this environment.
* **Preview**— intended to model the consumer application’s UAT environment. Preference is given to the version of content that is recognized as preview. If a preview version is not found, the production content will instead be visible.
* **QA**— intended to model the consumer application’s QA testing environment. Preference is given to the version of content that is recognized as QA. If a QA version is not found then the Preview version is considered, followed by the Production version. The first version that is found in the sequence is the version that will be used.
* **Development**— intended to model the consumer application’s development environment, including both the development server and local developer machines. Preference is given to the version of content that is recognized as development. If a development version is not found then the QA version is considered, followed by the Preview version and then the Production version. The first version that is found in the sequence is the version that will be used.

# Content Management System Environments

The content management system employs multiple environments to support its ongoing development. It is important to note that content development and consumption for consumers in all development phases will take place in the production environment; the alternate environments are intended only for ongoing CMS development. The primary driver behind this approach is to ensure that the consumer websites can evolve without any friction or needless coordination from ongoing CMS development. Another important factor is to ensure that consumers have access to the freshest content available for consumption. The following environments are employed:

* **Production**—the source of content for consumers, in all of their stages of development. The CMS production environment consists of the CMS Administration Portal, the primary content database, and a read-only content database which is kept in sync with the primary instance via SQL Server Replication. It is intended that content consumers use the primary instance for their production and UAT environments to ensure content freshness with no latency and use the read-only instance for their QA and development needs.
* **QA**—the environment responsible for allowing QA validation of ongoing CMS development. In addition to the QA version of the CMS Administration Portal, a copy of one of the current consumer websites will be bound to the in-development CMS client library and used as a vehicle to verify the CMS changes. Occasionally, when the CMS is evolving in parallel with a consumer, the actual consumer website will be pointed at the CMS QA environment, in order to allow new functionality from both the CMS and the website to be QA tested in unison. The CMS QA database should be refreshed from production at the beginning of each new QA cycle.
* **Development**—the primary target of ongoing CMS development, supporting both the development server and local development machines. In addition to the development version of the CMS Administration Portal, a copy of one of the current consumer websites will be bound to the in-development CMS client library and used as a vehicle to verify the CMS changes. Occasionally, when the CMS is evolving in parallel with a consumer, the actual consumer website will be pointed at the CMS development environment, in order to allow new functionality from both the CMS and the website to be developer-verified in unison. The CMS development database should be able to be refreshed from production on-demand.

# Sample Development Workflows

**Consumer Website Development**

1. During development, the consumer website continues using the current CMS client library version and uses the CMS Read-Only data store for its content source. Any CMS content for the new features is entered into the production CMS Administration Portal in its initial workflow status. The new content becomes visible after database replication takes place and the CMS cache is refreshed.
2. The website is deployed to the QA environment. It continues to use the CMS Read-Only data store for its content source. In the production CMS Administration Portal, any CMS content for the new features is approved and the status is updated to reflect its eligibility for QA. The new content becomes visible after database replication takes place and the CMS cache is refreshed.
3. The website is deployed to the preview environment. The content source is updated to point to the read/write production CMS data store. In the production CMS Administration Portal, any CMS content for the new features is approved and the status is updated to reflect its eligibility for Preview. The new content becomes visible when the CMS cache is refreshed.
4. The website is deployed to the production environment. The content source continues to point to the read/write production CMS data store. In the production CMS Administration Portal, any CMS content for the new features is approved and the status is updated to reflect its eligibility for Production. The new content becomes visible when the CMS cache is refreshed.

**Widget Development**

1. Widget content is entered into the production CMS Administration portal in its initial workflow status. The widget functionality is developed in the SiteFramework project. When the widget is ready for integration with the consumer website, the website updates its reference to the new SiteFramework library, continuing to use its current CMS client library version and the CMS Read-Only data store for its content source. The widget content becomes visible after database replication takes place and the CMS cache is refreshed.
2. The website, with new SiteFramework library, is deployed to the QA environment. It continues to use the CMS Read-Only data store for its content source. In the production CMS Administration Portal, the widget content is approved and the status is updated to reflect its eligibility for QA. The new content becomes visible after database replication takes place and the CMS cache is refreshed.
3. The website, with new SiteFramework library, is deployed to the preview environment. The content source is updated to point to the read/write production CMS data store. In the production CMS Administration Portal, the widget content for the new features is approved and the status is updated to reflect its eligibility for Preview. The new content becomes visible when the CMS cache is refreshed.
4. The website, with new SiteFramework library, is deployed to the production environment. The content source continues to point to the read/write production CMS data store. In the production CMS Administration Portal, the widget content for the new features is approved and the status is updated to reflect its eligibility for Production. The new content becomes visible when the CMS cache is refreshed.

**CMS Development**

1. The CMS changes, database and code, are made in the CMS development environment. When ready for integration testing, a copy of one of the current consumer websites is upgraded to the in-development CMS client library and used for verification. The website copy and CMS Administration portal use the CMS development data store for its content. The data store is refreshed from production as-needed to ensure parity between the environments.
2. The website copy, with new CMS client library, and CMS Administration Portal are deployed to the QA environment. They are updated to use the CMS QA data store for its content, which is refreshed from production as part of the deployment.
3. The CMS Administration portal is deployed to the production environment and any needed database changes are made to the production read/write and read-only instances. A new version of the CMS client library component is made available for consumers.

**CMS and Consumer Website Parallel Development**

1. The CMS changes, database and code, are made in the CMS development environment. The CMS Administration portal uses the CMS development data store for its content. The consumer website uses the current CMS client library version and uses the CMS Read-Only data store for its content source. Any CMS content for the new features is entered into the production CMS Administration Portal in its initial workflow status. The new content becomes visible after database replication takes place and the CMS cache is refreshed.
2. When ready for integration, the consumer website is upgraded to the in-development CMS client library and updated to use the CMS development data source, which is refreshed from production as needed to incorporate any new content development.
3. The website, with new CMS client library, and the CMS Administration portal are deployed to the QA environment. They are updated to use the CMS QA data store for their content, which is refreshed from production as part of the deployment. The CMS QA data store may be refreshed as needed to incorporate new content development.
4. The CMS Administration portal is deployed to the production environment and any needed database changes are made to the production read/write and read-only instances. A new version of the CMS client library component is made available for consumers. The website, with new CMS client library, is deployed to the preview environment. The content source is updated to point to the read/write production CMS data store. In the production CMS Administration Portal, any CMS content for the new features is approved and the status is updated to reflect its eligibility for Preview. The new content becomes visible when the CMS cache is refreshed.
5. The website is deployed to the production environment. The content source continues to point to the read/write production CMS data store. In the production CMS Administration Portal, any CMS content for the new features is approved and the status is updated to reflect its eligibility for Production. The new content becomes visible when the CMS cache is refreshed.