Content Staging with Kentico CMS 7

This white paper explores how organizations can leverage the Kentico CMS 7 content staging module to implement innovative solutions for managing content in a process centric manner that meets their unique needs.

Kentico CMS

TABLE OF CONTENTS

EXECUTIVE SUMMARY

SCOPE AND GOALS

It is the goal of this document to detail the features, design and best practices for utilizing the Kentico CMS content staging module.

This document focuses exclusively on content staging and does not provide an in-depth analysis of other Kentico CMS features or the Kentico CMS platform, except where necessary to provide context. After completing this white paper, the reader should understand:

- The purpose of content staging
- Example business scenarios for leveraging the technology
- The business case for implementing content staging
- Configuration of the Kentico CMS content staging module
- Architecture, design and implementation best practices

TARGET AUDIENCE

This white paper targets hands-on implementers of Kentico's content management system (CMS). The reader may be in a role such as:

- Administrator
- Site manager
- Developer

supporting the Kentico CMS platform. Readers should have general technical knowledge of the Kentico CMS platform. However, it is not necessary to have an in-depth understanding of content staging or the Kentico CMS content staging module.

While not specifically targeted at a non-technical audience, non-technical readers can use this white paper to understand the value of implementing content staging and get details for creating a business case.

WHAT IS CONTENT STAGING

Content staging is the practice of deploying changes to one or more servers outside of the development environment, such as content review, quality assurance, security validation or performance testing, before deploying to production. Despite the name, content staging is not just about staging visible content.

> ***Why is it necessary to separate staged changes? Changes applied to a staging environment could easily affect production.***

One of the key advantages of using a content management system is the separation of content from presentation and layout. Contributors create content in a display agnostic format and the system combines the content with templates, custom components and data to create a finished web page. This strategy speeds delivery and allows resources to perform the tasks which they are most suited. Content staging ideally addresses all aspects of managing content in a CMS – not just the content.

Content staging processes allow staff and/or test suites to verify changes and their impacts before applying them to a production site. Content staging processes may support content approval, review and manual or automated synchronization of changes among source and target servers.

In general, servers housing staged changes should be separate from production. In virtualized or cloud-based environments, "separate" does not necessarily imply physically separate hardware, but rather separation in terms of operational environments.

Why is it necessary to separate staging? Changes applied to items other than content in a staging environment with a live system could easily impact production: changes to shared components, high resource utilization during a load test or an ill-behaved custom component could have a dramatic negative impact on production or cause a complete outage. Additionally, if changes include more than content, organizations should consider staging the changes behind a firewall to minimize the risk of exposing security issues to the internet before validating their safety.

PEOPLE AND PROCESSES

THE VALUE OF CONTENT STAGING

There is no single model that works for all organizations for content staging. An organization that has one resource modifying and approving content with no developers has much different needs and control requirements than a large organization with departmental sites, custom code and site designers. Content management solutions like Kentico CMS provide business users the power and autonomy to manage their own content. It allows those who understand the business to manage their information assets, while simultaneously allowing technical and design resources to manage functionality and appearance. This model has a number of benefits, but decentralized content management also introduces a number of complexities.

Why stage content? Organizations have many reasons for wanting to stage changes prior to deploying to production:

- Does the content adhere to organizational compliance policies and standards?
- Is the content grammatically correct?
- Have accessibility requirements been taken into account?
- Has a custom component/code been tested?
- Does a new template or layout adhere to organizational or departmental design guidelines?
- Does the content duplicate existing content?

No matter what the specific reason, the value of a CMS solution is only as good as its underlying content. Governance processes and tools to systematically enforce them, like the Kentico CMS content staging module, enable organizations to avoid wasting resource time troubleshooting issues, reworking content or worse legal issues and fines related to compliance violations. Reviewing changes and ensuring they are correct prior to production deployment often takes much less time and effort – and definitely stress - than discovering a problem on a live site.

The Kentico CMS content staging module should not be used in isolation by technical resources. Although content staging is a tool, it is important that it be supported by users with well-defined responsibilities and a clearly understood process about how changes move between environments to production.

INTEGRATING CONTENT STAGING INTO GOVERNANCE PROCESSES

Many organizations have one or more existing governance process for managing applications, content, data or other organizational assets. Content staging should not be implemented in isolation. To the extent possible, when adopting content staging, site administrators or the resource sponsoring the adoption should make an effort to ensure use of the content staging module is documented as part of existing governance processes where applicable. For example:

EXISTING PROCESSES

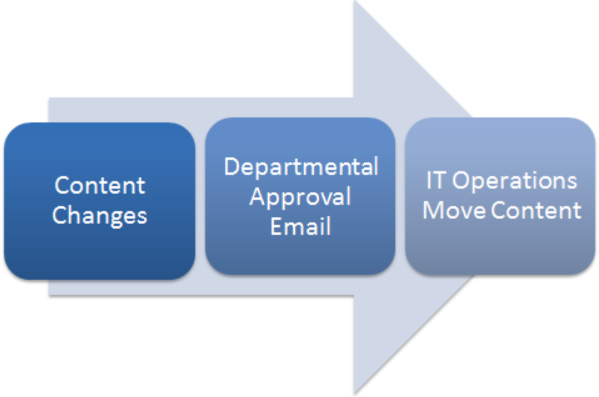


Figure 1 - Content Governance Before Content Staging Module

WITH CONTENT STAGING MODULE

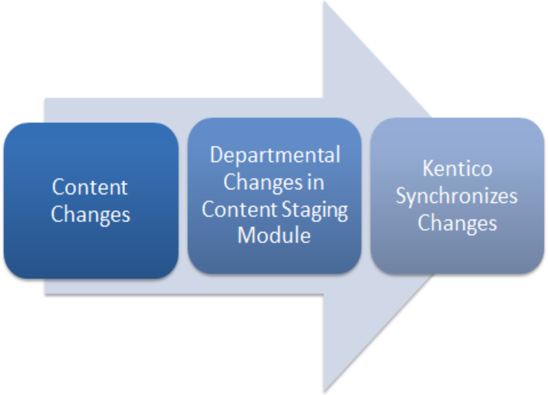

Content Changes
Departmental Changes in Content Staging Module
Kentico Synchronizes Changes

Figure 2 - Content Governance After Content Staging Module

In this case, introduction of the content staging module streamlines governance processes by providing an operational tool to perform content moves. Features of the content staging module may be integrated into multiple staging processes since content, data and application components/code may have different lifecycles.

DEFINING A NEW CONTENT GOVERNANCE PROCESS

For organizations that do not have an existing content governance process, introduction of the Kentico CMS content staging module may prompt a discussion of how different types of CMS content and objects move from their origin into the production environment.

The initial governance plan should have the simplest process flow possible to support the organizational needs. To develop a governance plan, form a small content governance committee that includes representatives from technology, the business and HR or legal to determine compliance requirements. Within the committee, ask questions such as:

- Who approves document changes?
- Can page owners redesign the page/site layout?
- Who can make changes to the overall branding for the portal?
- Who can manage objects such as document types and roles?
- Who controls the direction that content can be synchronized?
- How will the governance model be updated and maintained?

and create a Responsibility Assignment Matrix. Rows in the matrix represent individual responsibilities while specific columns represent a single role. Completion of the matrix requires assignment of one of four values to a corresponding role and responsibility:

- **R**esponsible - The role that actually accomplishes or executes a task.
- **A**ccountable - The role ultimately answerable for success or failure (R reports to A).
- **C**onsulted - Subject matter and technical experts brought in as needed.
- **I**nformed - People and departments who are kept apprised of findings, results and policies.

An example RACI is provided in the table below.

	ROLE 1	ROLE 2	ROLE 3	ROLE 4
Approve documents	A	C	I	R

Creating a new content staging process is likely to be a much larger effort than leveraging the content staging module. However, organizations should not try to do everything at once or design an overly cumbersome process that users attempt to circumvent.

THE KENTICO CMS APPROACH

Kentico CMS is a .Net-based content management system developed entirely with C#. Not just a tool for building web sites and portals, Kentico CMS provides a module-centric, open and extensible content framework that can speed delivery of dynamic sites and applications. Instead of focusing on creating infrastructure concerns, like security, leveraging Kentico CMS allows organizations to focus on the features that deliver business value.

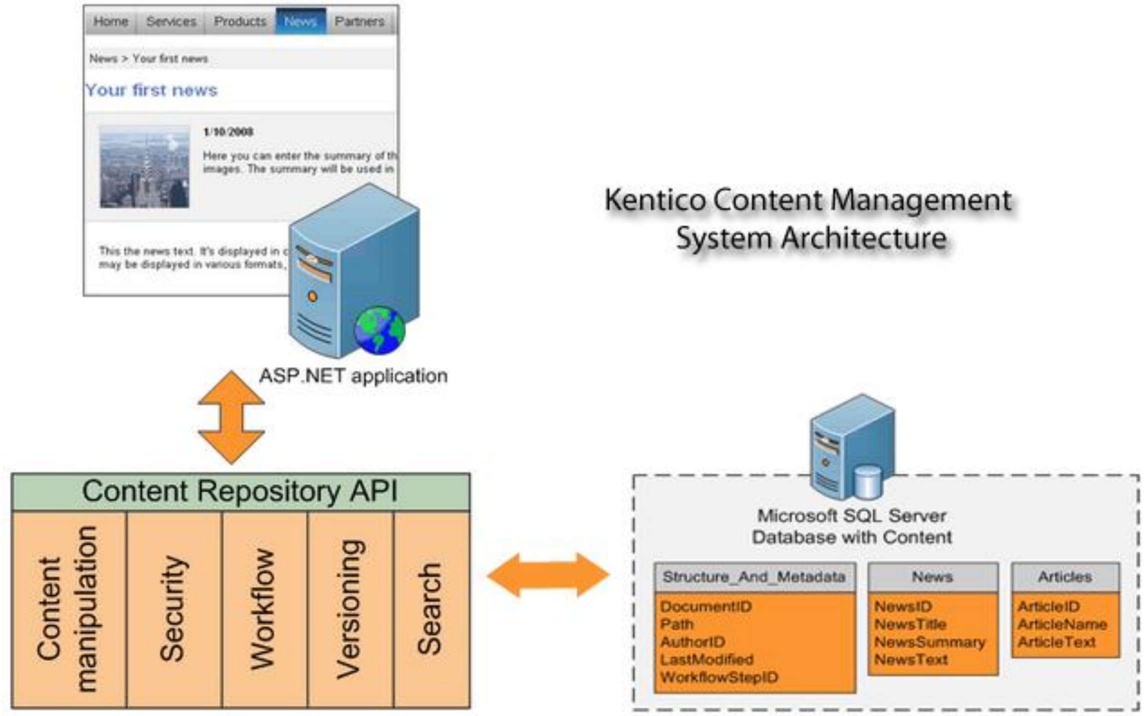


Figure 3 - Kentico Architecture

In addition to the core platform, Kentico CMS includes almost thirty pre-built modules that implement optional features like blogs, content rating, wiki and content staging. The pre-built modules developed by Kentico are distributed with the core platform and, unlike third-party components in some CMS platforms, provide users capabilities that look, feel and function identically to the platform core.

KENTICO CMS CONTENT STAGING MODULE

The Kentico CMS content staging module is one of many pre-built modules distributed with the platform core. The content staging module allows review and synchronization of changes, on either a manual or an automated basis (for documents), between servers. Users must define target and source servers to synchronize content. A source server is the server where changes originate and a target server is the destination for the changes. Kentico CMS allows users to define servers as both a target and source making it possible to bi-directionally synchronize changes.

Items synchronized by the content staging module can be organized into three categories.

STAGING CATEGORIES

DOCUMENTS

A document is one of the most fundamental units within Kentico CMS. Each document within the platform has a type, such as a page or news item, with its own:

- data structure
- editing form layout
- design
- queries

and other settings. Documents are one of the most common sources for synchronization, because organizations manage content even when they do not change objects and templates. The Kentico CMS content staging module captures the following CMS events:

TASK TYPE	DESCRIPTION
CREATEDOC	A document was created
UPDATEDOC	A document was modified
DELETEDOC	A document was deleted
DELETEALLCULTURES	All cultural versions of a document were deleted
PUBLISHDOC	A document was published
ARCHIVEDOC	A document was archived
REJECTDOC	A document was rejected
MOVEDOC	A document was moved to another area

Users can review and synchronize documents in the content tree, document attachments and document relationships using the content staging module.

DATA

The data category captures data changes to custom tables. The Kentico CMS custom-table module adds the ability to create user-defined tables within the Kentico CMS system database and manage the data via an intuitive user interface. The Kentico CMS content staging module captures:

TASK TYPE	DESCRIPTION
CREATEOBJ	A new item was added to the custom table
UPDATEOBJ	An item was updated in the custom table
DELETEOBJ	An item was deleted from the custom table

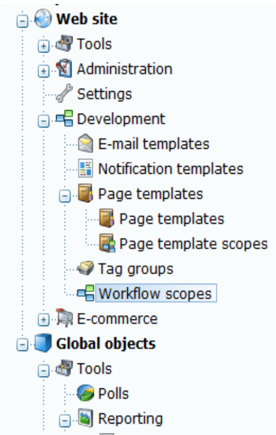


Figure 4 - content staging objects

Users with adequate permissions can review, synchronize or reject one or more changes to the custom tables.

OBJECTS

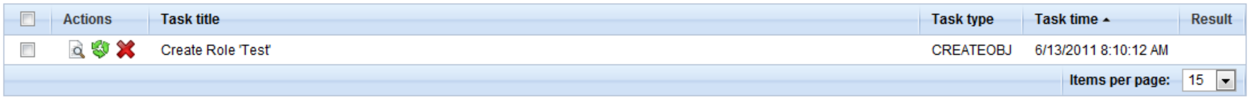
The object category is one of the most diverse areas synchronized by the content staging module. For example, changes to users, custom tables, workflow and media libraries are tracked and synchronized in this category. As illustrated in the figure on the left, items synchronized by content staging in the object category can be segmented into:

- website: changes connected to the current website
- global. changes for global objects.

The content staging module tracks the following events for activities performed in the user interface:

TASK TYPE	DESCRIPTION
CREATEOBJ	An object was created
UPDATEOBJ	An object was updated
DELETEOBJ	An object was deleted
ADDTOSITE	A site related object was assigned to the site
REMOVEFROMSITE	A site related object was removed from to the site
RENAMEFOLDER	A folder in the media library was renamed
COPYFOLDER	A folder in the media library was copied
MOVEFOLDER	A folder in the media library was moved
DELETEFOLDER	A folder in the media library was deleted

For example, if a new role "Test" is created, the content staging module would capture:



□	Actions	Task title	Task type	Task time ▴	Result
□		Create Role 'Test'	CREATEOBJ	6/13/2011 8:10:12 AM	

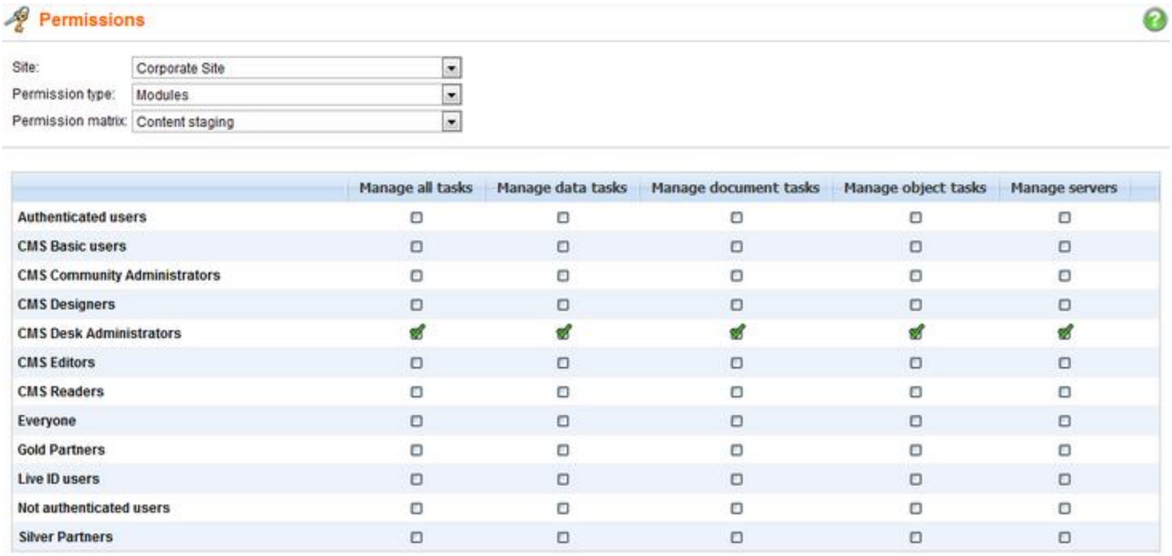
A complete list of synchronization items is available here.

SECURITY

Kentico CMS implements a number of security protections in the content staging module. Data transferred during synchronization occurs over a HTTPS secured SOAP-based web service. A definition of the web service, the WSDL, is located on the installation server at:

```
http://{hostname}/{install location}/ CMSPages/syncserver.asmx?wsdl
```

In addition, managing content staging tasks and performing synchronization requires a specific rights be assigned. Content staging is not a simple granted or denied permission. Kentico CMS allows granular control over content staging capabilities, which allows organizations the flexibility to configure the tool in a way that is closely aligned with their business processes.



Permissions

Site: Corporate Site
Permission type: Modules
Permission matrix: Content staging

	Manage all tasks	Manage data tasks	Manage document tasks	Manage object tasks	Manage servers
Authenticated users	□	□	□	□	□
CMS Basic users	□	□	□	□	□
CMS Community Administrators	□	□	□	□	□
CMS Designers	□	□	□	□	□
CMS Desk Administrators	☑	☑	☑	☑	☑
CMS Editors	□	□	□	□	□
CMS Readers	□	□	□	□	□
Everyone	□	□	□	□	□
Gold Partners	□	□	□	□	□
Live ID users	□	□	□	□	□
Not authenticated users	□	□	□	□	□
Silver Partners	□	□	□	□	□

Figure 5 - Content Staging Module Security Permissions

Administrators assign permissions to security roles to:

- Manage all tasks
- Manage data tasks
- Manage document tasks
- Manage object tasks
- Manage servers

Using these permissions, administrator could, for example, restrict synchronization of documents to a business department and synchronization of object to a development lead. Obviously, many other scenarios are possible.

PROGRAMMATIC ACCESS

In addition to the administrative interface for the content staging module, Kentico CMS provides access to synchronization using an API. All activities exposed in the administrative interface can be accessed programmatically to create customized processes that closely align with organizational needs.

Case Study: Programmatic Content Synchronization

Kentico CMS customer Jonas Software leveraged the Kentico CMS content staging API to create a custom synchronization process for their software-as-a-service offering. Many of Jonas' customers were frequently updating content while Jonas designers were simultaneously making changes to templates and other CMS objects such as custom document types. Jonas needed to keep content changes fluid, but exercise some governance over the structural aspects of the site.

Programmatic access to the content staging module provided the SaaS vendor with required flexibility to support the needs of customers and their internal design process.

Jonas created a custom user interface that allowed designers to synchronize layouts and templates from a local machine to a centralized site using the Kentico CMS content staging API. Creating the application simplified the design promotion process for designers and reduced the opportunity for user error on the centralized site. According to Randar Puust of Jonas Software,

> *"The technology was well documented and easy to use. It provided a mechanism for us to easily enforce our processes systematically without impacting customers."*

IMPLEMENTING CONTENT STAGING WITH KENTICO CMS

WHAT YOU CAN EXPECT

The Kentico CMS content staging module is an optional component that may be added during the installation of Kentico CMS or via a software update. Once the module is installed, three additional steps are required to use the content staging module. Rename the Microsoft.Web.Services3.dll.rename to Microsoft.Web.Services3.dll in the bin folder of the webserver where Kentico CMS is installed. This .dll provides the implementation of Microsoft's web services engine, which is used to synchronize content between servers. Next, users need to configure the source of item changes that will tracked by the content staging module. Users can specify the types of changes they wish for the content staging module to track, which may include:

- Content changes
- Data changes
- Object changes
- Synchronization tasks

Kentico CMS captures changes to all selected items as content staging tasks that can be synchronized to the target server. Finally, the target server is specified which will receive the staged changes. Tasks are only logged when at least one source and target server are defined.

Kentico CMS also supports both simple and advanced bi-directional synchronization of changes between servers as illustrated below:

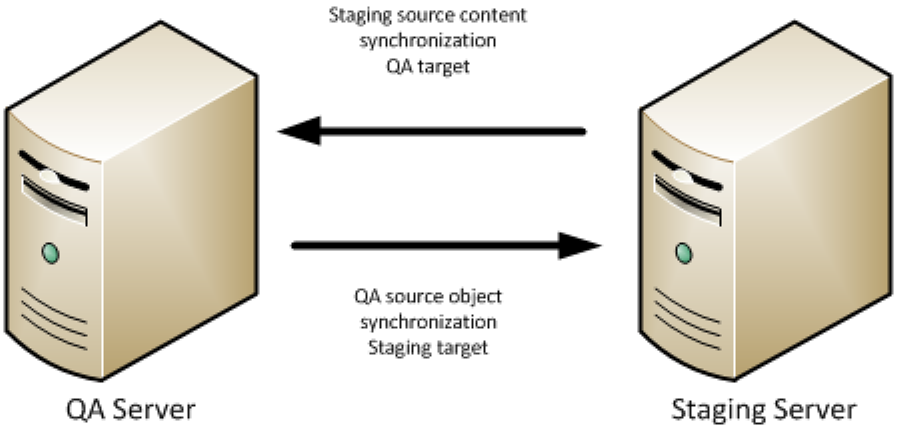


Figure 6 – Simple bi-directional staging configurations

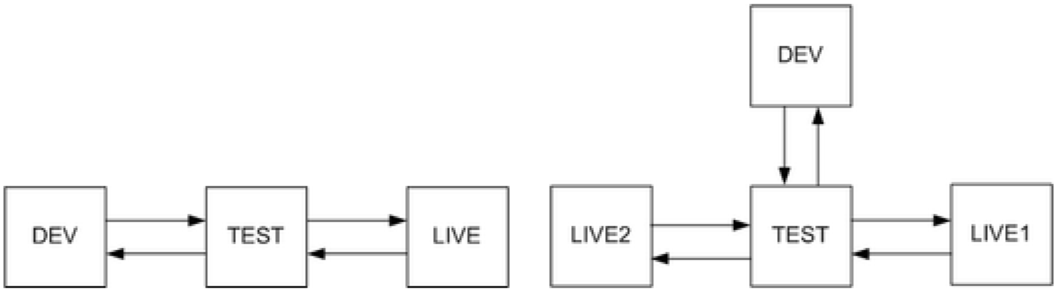


Figure 7 – Advanced bi-directional staging configurations

In this model each server is defined as a target and a source and changes flow between both. Special configuration is required to support bi-directional changes to avoid creating an infinite loop. The site administrator must disable the "log staging changes" option on each server involved in the bi-directional relationship or changes will be logged as new synchronization tasks on each server when synchronization occurs.

Additional details regarding configuration of the content staging module are available in the Kentico CMS Developer's Guide within the Content Staging Configuration topic.

ARCHITECTING CONTENT STAGING WITH KENTICO CMS

ARCHITECTURAL GUIDELINES AND BEST PRACTICES

SUPPORT MULTIPLE ITEM LIFECYCLES

Kentico CMS separates the process of content editing into distinct roles, which have different levels of rigor. Typically, it is not necessary for content changes to go through the same multi-stage processes as code and layout changes. Changes to custom code and layout may impact every page on the site, so it is important that they have a rigorous testing process to ensure there are no negative impacts.

Figure 7 represents on possible lifecycle model in which content changes are made in staging by contributors and promoted to production. Custom code changes go through a typical development lifecycle involving development, QA, staging which combines content changes with code and structural changes and finally production. It's important to remember that the physical files of web parts and form controls are not available for synchronization.

Although the lifecycles of content and the code are de-coupled, they are interdependent. Realistic content that is closely related to production should be used to develop and test code. The content relies on the code to define and display itself. There are multiple instances where these aspects get tangled. For example, when a new field is added to a document type it needs to be populated, sometimes manually, and the presentation templates need to be modified to display it. Figure 7 illustrates a content staging implementation where production is configured as a source server that pushes document changes to target servers custom code development and QA.

DEPLOY STAGING SERVERS BEHIND THE FIREWALL

As illustrated in figure 8, staging may be the first environment where document and other changes are combined to represent how the changes will look and behave in production. If changes other than content are being staged, it should occur inside of the firewall. Although code has been tested in a quality assurance environment, additional validation such a security audit or performance may take place in the staging environment where the content closely represents production.

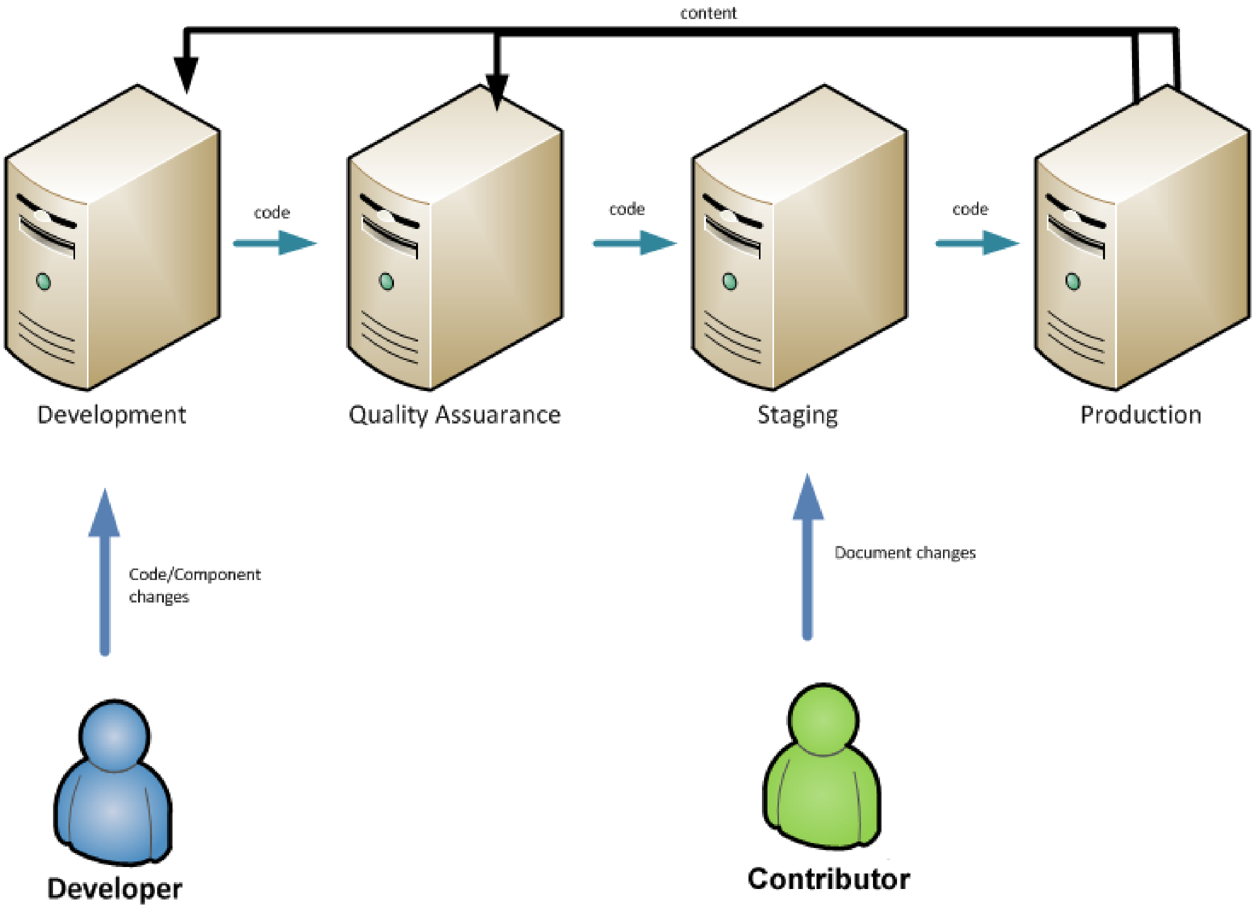
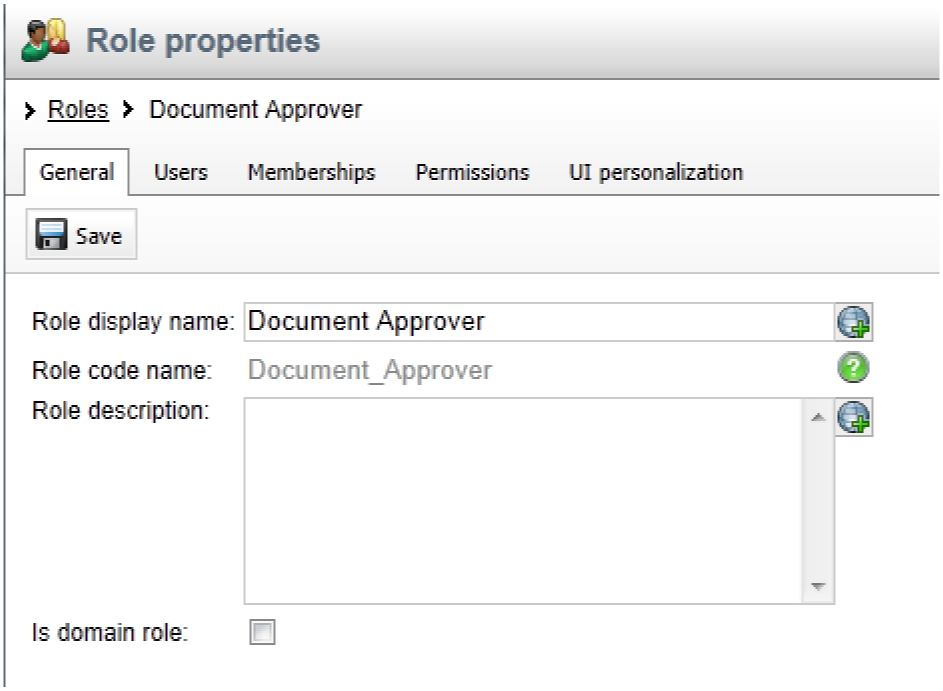


Figure 7 - Firewall placement

As you move left from production down to the development, code becomes less stable. Development may be a developer machine or a shared development environment containing changes that are untested beyond the unit level. Placing pre-production environments inside the firewall minimizes the risk of machine and network compromise due a security flaw or other issue being exposed to the internet. Furthermore, it provides an additional layer of protection to ensure that nobody outside of the corporate network can accidentally view staged content, which may not comply with corporate standards in the event security is improperly configured.

DEFINE CONTENT STAGING ROLES

Kentico CMS administrators should create specific content staging roles that are unrelated to organizational alignment and assign the roles to users that perform content staging and synchronization tasks. Creating new roles specifically for content staging is not only descriptive, but also allows content staging permissions to be added and removed without impacting other capabilities assigned to organizational roles. Figures 9 and 10 illustrate the creation of a new role and assignment of permission within the content staging module. A new role "Document Approver" is assigned the capability to manage document content tasks.



Role properties

› Roles › Document Approver

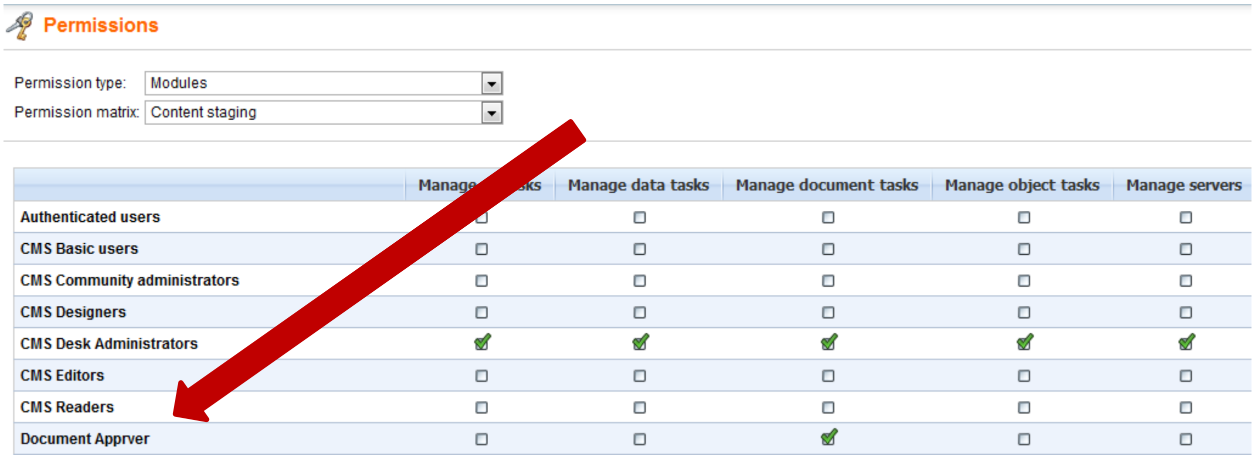
Role display name: Document Approver

Role code name: Document_Approver

Role description:

Is domain role: □

Figure 8 - Creating a new role



Permissions

Permission type: Modules

Permission matrix: Content staging

	Manage [illegible]sks	Manage data tasks	Manage document tasks	Manage object tasks	Manage servers
Authenticated users	□	□	□	□	□
CMS Basic users	□	□	□	□	□
CMS Community administrators	□	□	□	□	□
CMS Designers	□	□	□	□	□
CMS Desk Administrators	✓	✓	✓	✓	✓
CMS Editors	□	□	□	□	□
CMS Readers	□	□	□	□	□
Document Apprver	□	□	✓	□	□

Figure 9 - Content staging permissions

UNDERSTAND SYNCHRONIZATION VOLUME AND PERFORMANCE IMPACT

Administrators should profile synchronization activities outside of production prior to utilization on the live site to understand the maximum content sizes that can be synchronized to the production environment without negatively impacting performance.

Additionally, some synchronization activities, like media libraries, support defining a maximum object size. Objects larger than the user-defined maximum in kilobytes are not synchronized by the content staging module. This setting minimizes the risk that production performance will be impacted due to synchronization activities. Administrators can configure Kentico CMS to restrict synchronization of media for large by adding

```
<add key="CMSMediaFileMaxStagingSize" value="1024" />
```

to the application settings (appSettings) section of the web configuration (web.config) for the Kentico CMS ASP.net application. The value is the maximum file size in kilobytes.

PROVIDE EACH ROLE THE MINIMUM SET OF PERMISSIONS NECESSARY

Do not assign user roles more permissions than are required to perform content staging tasks. This is a general security best practice that reduces the threat surface area in the event that user account is compromised or the role is assigned incorrectly.

OTHER OPTIONS FOR CONTENT SYNCHRONIZATION

SITE IMPORT AND EXPORT

Kentico CMS also supports exporting data from one instance and importing it into another one to synchronize content. However, this method has a number of limitations as compared to content staging. When using the site import/export feature to import documents to an existing site, only new documents are imported and created - existing documents are not updated. If it is necessary to update documents, users must remove the site or import it as a new package or leverage content staging, which supports update events. Additionally, it is not possible to leverage the export/import feature to make changes to a single document. The feature is all or nothing. Synchronization of a single document requires use of the content staging module.

WORKFLOW

Kentico CMS customer, Shaw Industries, uses the content staging module and Kentico CMS workflow functionality to approve and synchronize CMS items. Dwayne Gould of Shaw Industries said his team wanted to provide maximum flexibility for departmental users so they implemented a custom solution for content staging leveraging the content staging API with workflow.

Kentico CMS is designed for ease of use, but also to ensure developers have as much access as necessary to the underlying ASP.Net environment. Kentico CMS provides developers complete programmatic access to workflow and content staging components allowing the creation of sophisticated solutions to support internal and external users.

CONCLUSION

The Kentico CMS content staging module provides a flexible solution for allowing users and administrators to review and synchronize changes between Kentico CMS servers. The out of the box implementation supports granular configuration of:

- Content to be synchronized
- Security permissions
- Content sources and targets

via the administrative user interface. If, however, the supported use cases do not meet organizational needs, Kentico CMS empowers developers to create custom solutions by exposing the functionality programmatically and providing complete API documentation.

ABOUT KENTICO

Kentico is a global provider of web content management solutions built with the Microsoft .NET platform. The Kentico CMS platform is used by small and large organizations in over 80 countries. The company's flagship product is Kentico CMS for ASP.NET. The Kentico CMS platform allows clients to build and manage a wide variety of website and web application solutions from corporate websites to Web 2.0 properties and collaboration portals.

Ease of use is a core concern for Kentico. The company has created a platform that allows users to quickly create a site using module configuration via a web-based interface instead developing custom code from scratch, but augment the of the box solution via its extensive API when necessary.