Building Multilingual Solutions Using 2007 SharePoint Products and Technologies

# Introduction

This paper provides information and guidelines in using Microsoft Office SharePoint Server 2007 on scenarios where content is managed across different languages. It details most of the functionality provided by Microsoft Office SharePoint Server 2007 and suggests how to manage them in multi lingual scenarios. It also provides examples for creating custom web parts using Microsoft Visual Studio 2005 and how to configure and manage them using SharePoint Designer 2007, next generation of web designer tools specifically designed for SharePoint web sites. Windows SharePoint Services V3 features and functionalities are also described to add value to the end user experience. Finally, this paper came with a full functional example that, using all the features described in this whitepaper, provides a template to create a multilingual Internet site.

Microsoft Office SharePoint Server 2007 provides enterprise-scale capabilities to meet business-critical needs like managing content and business processes, simplifying how people find and share information across boundaries, and enabling better informed decisions. Microsoft Office SharePoint Server 2007 supports all of the intranets, extranets, and Web applications across an enterprise within one integrated platform, instead of relying on separate fragmented systems.

Microsoft Office SharePoint Server 2007 can offer solutions in the following areas:

**Provide a simple, familiar, and consistent user experience:** Microsoft Office SharePoint Server 2007 is tightly integrated with familiar client desktop applications, e-mail, and Web browsers to provide a consistent user experience that simplifies how people interact with content, processes, and business data. This tight integration, coupled with robust out-of-the-box functionality, helps you employ services themselves and facilitates product adoption.

**Help meet regulatory requirements through comprehensive control over content:** By specifying security settings, storage policies, auditing policies, and expiration actions for business records in accordance with compliance regulations, you can ensure your sensitive business information can be controlled and managed effectively. And you can reduce litigation risk for your organization. Tight integration of Microsoft Office SharePoint Server 2007 with familiar desktop applications means that policy settings are rendered onto client applications in the Microsoft Office system, making it simpler for employees to be aware of and comply with regulatory requirements.

**Effectively manage and repurpose content to gain increased business value:** Business users and content authors can create and submit content for approval and scheduled deployment to intranet or Internet sites. Managing multilingual content is simplified through new document library templates that are specifically designed to maintain a relationship between the original version and different translations of a document.

**Simplify organization-wide access to both structured and unstructured information across disparate systems:** Give your users’ access to business data in common line-of-business systems like SAP and Siebel through out-of-the-box connectors in Microsoft Office SharePoint Server 2007. Users can also create personalized views and interactions with business systems through a browser by dragging and dropping predefined, configurable back-end connections. Enterprise-wide Managed Document Repositories help your organizations store and organize business documents in one central location.

**Connect people with information and expertise:** Office SharePoint Enterprise Search incorporates business data along with information about documents, people, and Web pages to produce comprehensive, relevant results. Features like duplicate collapsing, spelling correction, and alerts improve the relevance of the results so you can easily find what you need.

**Share business data without divulging sensitive information:** Give your employees access to real-time, interactive Microsoft Office Excel 2007 spreadsheets from a Web browser through Excel Services running on Microsoft Office SharePoint Server 2007. Use these spreadsheets to maintain and efficiently share one central and up-to-date version while helping to protect any proprietary information embedded in the documents (such as financial models).

**Enable better-informed decisions by presenting business-critical information in one central location:** Microsoft Office SharePoint Server 2007 makes it easy to create live, interactive business intelligence (BI) portals that assemble and display business-critical information from disparate sources, using integrated BI capabilities such as dashboards, Web Parts, scorecards, key performance indicators (KPIs), and business data connectivity technologies. Centralized Report Center sites give users a single place for locating the latest reports, spreadsheets, or KPIs.

**Provide a single, integrated platform to manage Intranet, Extranet, and Internet applications across the enterprise:** Microsoft Office SharePoint Server 2007 is built on an open, scalable architecture, with support for Web services and interoperability standards including XML and Simple Object Access Protocol (SOAP). The server has rich, open application programming interfaces (APIs) and event handlers for lists and documents. These features provide integration with existing systems and the flexibility to incorporate new non-Microsoft IT investments.

## Requirements and Keywords

Readers of this paper should be familiar with the following technologies:

* ASP.NET 2.0
* Microsoft Visual Studio 2005
* Microsoft Office SharePoint Server 2007
* Microsoft Windows SharePoint Services 3.0
* Microsoft SharePoint Designer 2007

In this paper the following words reference on the following concepts:

**Server Farm:** A central group of network servers maintained by an enterprise. A server farm provides a network with load balancing, scalability, and fault tolerance. In some configurations, multiple servers may appear to users as a single resource. Each SharePoint farm has a single, unique Configuration Database where information and configurations of farm are registered. Each Server in the farm relies on that Configuration Datbase to get information about the farm and to provide services in the farm.

**Web Application:** A virtual server that resides on an HTTP server but appears to the user as a separate HTTP server. Several web applications can reside on one computer, each capable of running its own programs and each with individualized access to input and peripheral devices. Each web application can have its own domain name and IP address.

**Site Collection:** A set of Web sites on a web application that have the same owner and share administration settings. Each site collection contains a top-level Web site and can contain one or more sites (or sub-sites). There can be multiple site collections on each web application. A site collection can use only a single content database. Everything is now a site collection: a portal is a site collection where “Home” is the top-level web site and the areas are sites (or sub-sites), a channel hierarchy is a site collection where the root channel is the top-level web site and sub-channels are sites (or sub-sites).

**Top-Level Web Site**: The top, root default site in a site collection. Every Site collection has, on its top, a Top-Level Web Site. Access to the top-level Web site is provided supplying the URL of the site collection (like <http://ServerURL> or <http://ServerURL/sites/SiteCollectionName>) without specifying a page name or sub-site.

**Site (or sub-site):** A complete Web site stored in a named leaf of the top-level Web site. Each sub-site can have administration, authoring, and browsing permissions that are independent from the top-level Web site and other sub-sites.

**Publishing Pages:** Publishing pages are pages under “/Pages/” document libraries on each site with specific content types. Content for these pages are stored in this document library like columns in a list, with each column storing data for a page field.

**Collaboration Content:** Collaboration content is content stored in lists like calendars, lists, task lists, document libraries.

## What a Multilingual Scenario Is

In multilingual scenarios, information and content are provided in many languages, and users consume information in their preferred language. Multilingual features might be required by corporations that have subsidiaries in many countries and want to provide information to each country in its specific language, or by customers whose business spans many countries. Most of multilingual implementations rely on different environments: content editors need to maintain content updated on every language and IT administrators need a way to do it safely and securely. Getting contents from one environment, translate it and put it in final environment are, most of the time, activities that require lot of time and efforts. Writing custom solutions to obtain the same result may require a lot of effort too and resources, and efforts can grow if the number of languages to manage increases. In these scenarios, a good multilingual solution can provide server consolidation and operational costs reduction with the ability to host and keep in sync different web applications in different languages on one single environment and providing self provisioning features for auto-create sites in different languages for.

Multilingual as word per se can also have different means like:

* To manage content in different languages
* To navigate an Internet site or a Corporate portal in my preferred language
* To collaborate with people using different areas in different languages on the same application
* To manage and administer my spaces using my preferred language
* To search and browse content across my company using my preferred language

Microsoft Office SharePoint Server 2007 can provide the infrastructure to address and manage requirements described above with new features built to help users in creating and having a great multilingual experience:

* With the a new feature called “*Variation*” it is now easy to keep content in sync between different sites
* With the new workflow capabilities it is now possible to add translation process to sites
* Content can now be exported from a site into packages and provided externally for an easy translation process
* Users can now be redirected to a specific site based on their language preferences
* With Microsoft Office SharePoint Server 2007 built on top Windows SharePoint Services V3 users can host different sites in different language or local in the same site collection
* With full Unicode support user can have text (like title, column names, column values) in different languages; moreover having different documents in different languages on same document library is also supported.

## What This Whitepaper Covers

Main goal of this whitepaper is to introduce some key multilingual scenarios and to describe features provided by Microsoft Office SharePoint Server 2007 to help users with these scenarios. Information provided by this whitepaper will cover:

* Language Template Packs
* Variation
* Search
* Content Deployment
* Web Parts Customization

# Scenarios

## Multilingual User Experience

Microsoft Office SharePoint Server 2007 provides capabilities to create, manage and read content, but it also provides tools to administer sites, site collections and farms. Multilingual experience can be different depending on user rights and on task. This list describes multilingual experience provided by Microsoft Office SharePoint Server 2007:

|  |  |
| --- | --- |
| **Activity** | **Multilingual Experience** |
| **Read and manage content, site navigation, search** | Full multilingual experience OOB except for search where customization is needed |
| **Site administration** | Full multilingual experience OOB |
| **Site Collection administration** | Single language experience depending on the top level web site language |
| **Farm administration** | Single language experience depending on SharePoint installation language |

The following language considerations are applied to a Microsoft Office SharePoint Server 2007 server farm:

* MOSS 20007 supports one installation language per server farm. The same language version of SharePoint Portal Server must be installed on all servers in the server farm.
* Language Template Packs can be installed on server farm, however all servers in the farm must have the same Language Template Packs installed.
* Site collections and sites can use different languages depending on language packs installed
* Site collections can contain sub sites in different languages

## Collaboration Team Sites

In this scenario users have a place where they collaborate by creating content and sharing information. This scenario is typical composed by a small number of users who understand a common language. Normally, this scenario doesn’t require content translation and the scope is usually a single site. Those sites can be single island of information or can be grouped under a Corporate Portal. Both Windows SharePoint Services V3 and Microsoft Office SharePoint Server 2007 can fit in this scenario using features like language template packs, search and self service site creation. Users can create site collections in different languages by themselves, without relying on farm administrators. Moreover, site collection owners can also create sub sites in different languages in the same site collection. These concepts can now be applied to both Collaboration and Internet portals where users are able to create sub sites in the portal’s hierarchy in different languages.

## Corporate Enterprise Collaboration Portals

In this scenario a single company or an enterprise has a portal where information is available in different languages. This portal can be located on a single place or geographically distributed (Geo-deployment) in different locations. Information needs to be managed and reached by employees using their preferred languages. Depending on requirements, the entire content can be provided in different languages or only a specific sub set of content can have this requirement (like company news). Employees need also a personal space where store information and where collaborate each other in their preferred languages. Most features provided OOB by Microsoft Office SharePoint Server 2007 can help people to implement this scenario. A new Microsoft Office SharePoint Server 2007’s feature called “Variation” provides the ability to keep publishing pages among sites in a site collection synchronized. Moreover, the Variation feature can redirect users to the correct site based on their language preferences. Search can be tuned to provide search experience to user in their preferred language; doing this, users will be able to submit queries or look for content using a familiar user interface. Finally in some specific scenarios, using a new Microsoft Office SharePoint Server 2007’s feature called *Content Deployment*, the same portal can be replicated in read-only instances on different location to provide a fast and better experience to users across boundaries. This paper provides, in sections below, information some generic guidelines on using Content Deployment feature for multilingual geo-deployment scenarios.

## Internet Portal

In this scenario companies need to create an Internet Portal to publish content in Internet. Content has to be published in many languages so internet users are able to read and look for content in their preferred languages, or they can switch between different languages for the same content, having a *dynamic* multilingual user experience. On the other hand content owners need a tool that provides a way to create and keep content synchronized in different languages. They also have to provide content to 3rd party for translation safely, without letting external users access the staging environment. Content approval workflow features, to ensure that only approved content is online is most of the time a requirement too. Finally, this portal can be located on a single place or geo-distributed across boundaries to provide a better experience on performances to users. Just like the above scenario, features provided OOB by Microsoft Office SharePoint Server 2007 can also help people to implement this scenario too. Using the Variation feature content can be kept synchronized in different languages (on different sites) and users can be redirected to their preferred languages. Search can also be configured to provide search experience to user in their preferred language; and, if necessary, using the Content Deployment feature, the same portal can be replicated on read-only instances to different locations to provide a fast and better experience to users across boundaries.

## Enterprise Search Portal

In this scenario a single company has different portals in different languages where users manage and share different content in their preferred language. This is, for instance, a typical scenario in enterprises where each subsidiary needs a single divisional portal for local content. However, often there’s a need to have a single centralized place where users can land to perform search on all content of the company. Using enterprise portal search, users are then able to find specific content among all divisional portals using a friendly UI that provides search options in their preferred languages. Microsoft Office SharePoint Server 2007 provides feature to create a solution for this scenario: by having different site collections in different languages on the same farm it is now possible to create different divisional portals in different languages using the same hardware infrastructure. With The Variation feature it is also possible to set up an enterprise search portal that provides search options on user’ preferred languages. Topologies and farm configurations in Microsoft Office SharePoint Server 2007 provide options to share services between site collections in different languages on the same farm or among different farms.

# Get Deeper On Features

## Language Template Packs

Language template packs allow site owners to create SharePoint sites and site collections in multiple languages allowing them to manage content for each language in different sites. User can create a site or a site collection based on a language-specific site template by choosing the Language-Country ID (LCID). This language ID determines the language that is used to display and interpret text that is input on the site or site collection. For example, a site in French has the site's toolbars, navigation bars, lists, and column headings appear in French. A site in Arabic has the site's toolbars, navigation bars, lists, and column headings appear in Arabic. In addition, the default left-to-right orientation of the site changes to a right-to-left orientation to properly display Arabic text. There are two different editions of language template packs:

* **SharePoint Language Pack:** These are language packs designed for Windows SharePoint Services V3 configuration only. In a server farm configuration the same version and language of SharePoint Language Packs have to be installed on every machine of the farm
* **Server Language Pack:** These are language packs designed for Microsoft Office SharePoint Server 2007 configuration because of the number of extra templates included in Microsoft Office SharePoint Server 2007. In a server farm configuration the same version and language of Server Language Packs have to be installed on every machine of the farm. These language packs cover Windows SharePoint Services template also. If you want to create a site collection or a site using Windows SharePoint Services V3 template, you are able to select a different language without installing any additional SharePoint Language Pack because Server Language Pack includes SharePoint Language Pack also.

## Variation

The Variation feature in Microsoft Office SharePoint Server 2007 enables the capability of linking content in different sites within a site collection; when a publishing page is created in one site which is defined as “the source of Variation” and approved, a corresponding page is automatically created in all the target sites who participate in the Variation hierarchy with the same content and, optionally, using a different layout. Variation simply creates multiple sites with the same content presented in different formats, layout and or language. The format might change to simpler pages for mobile devices, different chrome for intranet, extranet and Internet users, or multiple languages for International corporations.

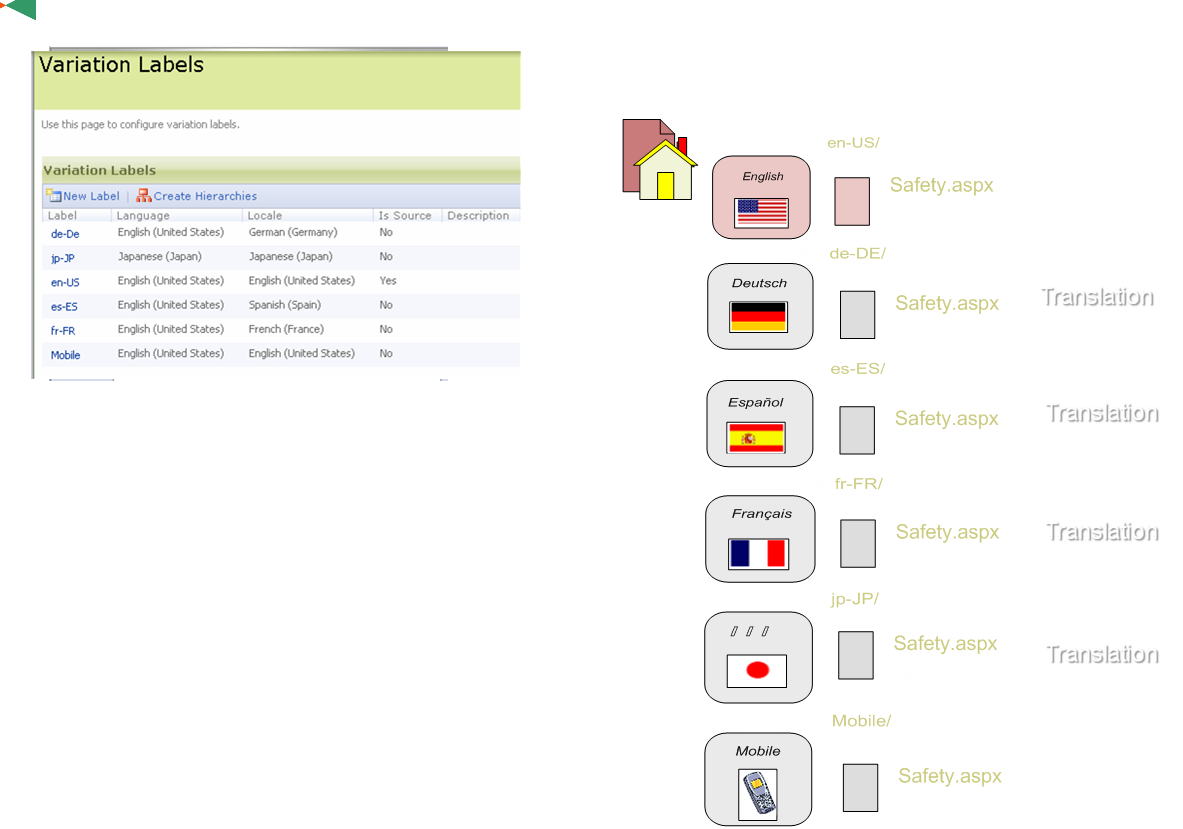


Figure 1: Variation’ Labels in Variation’s Hierarchy

### Understanding Variation

The Variations feature “works” within a single site collection. Setting up Variation feature means:

**Connect Variation’s Home:** Choosing the home of Variation means defining which content will be synchronized by Variation in the site hierarchy. There are, basically, two options: promote top-level web site as Variation’s home or select a different sub-site. Once defined the Home, user will never be able to hit the Home’s site because Variation will always redirect the user in his “preferred Variation’s label”, that means: a Variation target site who matches his preferred language. In an Internet Portal scenario, Variation’s Home is normally the at top-level web site of a site collection because all the content has to be proposed in different languages. On the other hand, a collaboration Portal can have only a specific sub-site (like company news) where the content is proposed in different languages.

**Define Variation Hierarchy:** Variation’s hierarchy is the skeleton of the Variation feature. By defining hierarchy a user chooses how many languages to support in his environment. Each language is identified as a “label” which is a specific site with a specific language. This is not a one-time operation. User can add or remove labels in the hierarchy at any time.

**Choose the Source of a Variation:** When building a Variation’s hierarchy it is mandatory to identify one, and only one, of the labels as the “source of Variation”. Source label is special: Variation will always propagate new content and updates from source site to all target sites in Variation’s hierarchy. Variation will also create target sites, in building the hierarchy, based on the site template configured in the source.

**Define Content Format in Variation Target Sites:** Optionally users can define different format and chrome for target sites. Imagine, for instance to have, in your variation’s hierarchy a label in Arabic. Arabic content is displayed as a mirror of English content: if the search box is on page in an English site, for example, at top left of the page, the same page, in Arabic, has to show the same search box on top right. Site administrators can choose different page layouts and different master pages for Variation’ target sites. There are different places and options to do that:

* *Master pages* are defined at target site level. To change master page on a specific target site, simply change master page setting as sub-site level once the hierarchy has been created
* *Page layouts* settings are provided in master page and page layout gallery, simply editing page properties of each layout page.

**Understand and Plan Content Translation:** There are two options for content owners to translate content on target sites: they can translate the content in the site using the web content editor tool or they can export content to a Variation’s package. This second option is quite useful, for instance, if a company relies on 3rd party for content translation: instead of let external user access staging environment directly, content can be exported in a package and provided to translators. Once translated, the content can then be re-imported and approved by a site manager.

**Understand Redirection Logic:** Once a Variation hierarchy is created, The Variation feature replaces the default page of The Variation’s Home site with a special page called *VariationRoot.aspx*. Every time an user hits the Variation’s Home, *VariationRoot.aspx* by default redirects them to a Variation’s target site based on browser language’ settings. For instance, if a users browser’ default language is set up to use French language, *VariationRoot.aspx* will redirect that user to French target site (assuming a French Variation’s label is in the hierarchy). This behavior can be customized by replacing *VariationRoot.aspx* page with a different page. This new page, for instance, can implement additional business logic in identifying user’s preferred languages. The following article describes how to customize Variation’s Redirection Logic:

*How to: Customize the Variation Root Landing Logic****:***

<http://msdn2.microsoft.com/en-us/library/ms562040.aspx>

**Variation’s Navigation Experience**: Variation exposes a control to let user navigate through labels. This control is called “Variation Label Control” and it is, by default, positioned on top of all master pages where “Publishing feature” is enabled. This control can be customized to provide different navigation experiences through Variation’ labels. Users can also customize Navigation Logic by simply modifying the *LabelMenuConfiguration* attribute of *VariationHierarchicalDataSource* with one of the following values:

* *LabelMenuConfiguration= 1:* means that user is redirected exactly to the same page on selected Variation’s label (if exists). For example from <http://MyInternetPortal/EN/SubSite/Index.aspx> selecting “Japanese” Variation’s label user is redirect to <http://MyInternetPortal/JP/SubSite/Index.aspx>
* *LabelMenuConfiguration= 2:* means that user is redirected to the default page on the same path on the selected Variation’s label. For example from <http://MyInternetPortal/EN/SubSite/Index.aspx> selecting “Japanese” Variation’s label user is redirect to <http://MyInternetPortal/JP/SubSite/default.aspx>
* *LabelMenuConfiguration= 3:* means that user is redirected to the default page of the selected Variation’s label. For example from <http://MyInternetPortal/EN/SubSite/Index.aspx> selecting “Japanese” Variation’s label user is redirect to <http://MyInternetPortal/JP/default.aspx>

By default, VariationLabelMenu control is configured as following:

|  |
| --- |
| <%@ Control Language="C#" %> |
| <%@Assembly Name="Microsoft.SharePoint.Publishing, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"%> |
| <%@Register TagPrefix="CMS" Assembly="Microsoft.SharePoint.Publishing, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" namespace="Microsoft.SharePoint.Publishing.WebControls"%> |
| <!-- Begin VariationLabelMenu --> |
| <cms:VariationsLabelEcbMenu id ="varlabelmenu1" DataSourceID="LabelMenuDataSource" DisplayText="<%$Resources:cms,VariationLabelMenuTitle%>" runat="server" /> |
| <cms:VariationHierarchicalDataSource id="LabelMenuDataSource" LabelMenuConfiguration="1" runat="server"/> |
| <!-- End VariationLabelMenu --> |

The following article describes how to customize Variation Label Control:

*How to: Customize the Variation Label Control Logic****:***

<http://msdn2.microsoft.com/en-us/library/ms551625.aspx>

### Variation’s Process

This summary provides a quick walk through the steps of the Variation process.

**New Content is Created on Source Site:** A content editor creates a new page or a new version of an existing page under “/Pages/” document library in Source site. By default once a new page (or version) is created, it has to be approved and published before it “goes live". Publishing the page is the “kick-off” action for the Variation process: once a page has been published Variation is now able to copy it through Variation’s hierarchy.

**Variation’s Job at Work:** By default, every 20 seconds Variation job checks for a published pages. If it founds a new one, it copies that page to all target sites in the Variation hierarchy. By default, copied pages on target sites need to be approved by local content editor before “go lives”. This is a one way pushing process from Variation source site to all target sites in the Variation hierarchy; also, external content on the page is re-configured based on Variation’s settings during copy process.

**Pages Have been Copied Through Hierarchy:** on each target site there’s now a new page created by Variation’s job as copy of the one on Variation’s source site. This page, at this stage, has the following characteristics:

* *Page settings and controls language:* page’ controls and settings reflect the language of the label. If, for instance, a label is in Japanese language all controls (like editor controls) or settings (like web part settings, page settings, editor toolbar) in target site will be in Japanese language
* *Page’s master page*: master page of page is inherited from master page of the site. By default, all target sites use same master page. However, these settings can be changed by administrators.
* *Page’s layout page*: by default pages in all target sites use the same layout page. However, administrators can set up a preferred layout page for one or more labels changing settings in Pages Layout.
* *Page’s content:* that’s the part of the page that has been copied as is through all target sites in the hierarchy. For example: if my source site is in English language and I’ve created a new page with English content, content will still be in English language on all copied pages in all target sites in hierarchy once Variation’s job is over.

**Now, Translate Your Content:** Editors and translators must now get involved; as mentioned before, content owners have a couple of options to do this task:

* They can choose to let translators connect to their staging environments on target sites and translate content on pages using web UI interface.
* They can export content, provide exported content to translators, let translators translate the content and, finally, re-import translated content on sites. During export process on a Variation target site, all pages are exported and packaged; that might mean a lot of content. A feature called *Translatable columns* can help translators in identifying what they need to translate. This setting defines which of the content in a package has to be translated simply creating a file in the package that identifies that content. A translator has to open this file in the package to identify which content he has to translate and which not.

**Last Step:** Page is now ready to be approved and published. That’s the last step required before a page on a target site is “live” to everyone. Additional workflows or other (like event handlers) can be added to “/Pages/” document library in Variation’s target sites to add additional logic to translation process.

### Constraints and Rules

In configuring the Variation feature is important to keep in mind the following constraints and rules:

**One Hierarchy per Site Collection:** Variation feature is available at site collection level where only one Variation’s hierarchy can exist. There might be scenarios where cross site collection copy is required. However, to address this scenario some custom code / logic is required.

**One To Many:** Variation’s process, when happens, always starts from Variation’s Source to all Variation targets in the hierarchy. This is a one to many process where the “master content” is always created and edited in source Variation site.

**Not for all Kind of Content:** Variation feature threats, basically, publishing pages and content referenced on publishing pages using the following logic:

* *Linked content* are threat by Variation depending on Variation’s settings but, in order to let Variation’s job copies that content on Variation’ target sites, “containers” have to exists before. For example. If you set up the Variation to copy resources on target sites during copy process and you put a link in a page to a document who is in “My Personal Documents” custom document library created on the Variation source site, “My Personal Documents” document library needs to exist on all Variation’ target sites before Variation job runs to let Variation’s job copies referenced document on all target sites.
* *Sub-sites under Variation source site* need to have “Publishing Feature” enabled in order to let Variation copies them through all Variation’ target sites.
* *All publishing pages* are always kept in sync through all target sites by Variation’s job. To avoid that a site admin on a target site can make a local copy of the page and delete the original page. This will break the hierarchy for that page in that target site.

**Be Aware of Web Parts:** By default web parts are updated on pages on target sites by Variation’s job. While this is a good option in most of cases, it might be a problem for web parts that are not designed to be aware of Variation feature. A good example are Windows SharePoint Services’ list web parts: those web parts always reference lists using site level GUID identifier and, because they don’t know anything about Variation’s feature and how it works, putting these web parts on a publishing page and configure Variation to keep them synchronized through target sites will cause an error in rendering that page on a target site after Variation process is done. To avoid this behavior there’s a settings on Variation that prevent Variation’s job to threat web parts on publishing pages. As result, if there are web parts on pages on source site, those web parts will never be copied on pages on target sites during Variation’s process. However, custom web parts can be wrote to be “Variation aware” so, they can have logic to set up information and content in the web part during Variation’s process. See “Custom Web Parts” paragraph later in this paper for additional information on web parts and Variation feature.

**New Versions vs. Mistakes:** Every time a Variation’s process occurs, it copies the entire changed publishing page from source site to all target sites in Variation’s hierarchy. If editor on Variation’s source target needs to fix an orthographic error or a typo he did on previous version there is no way, out of the box, to communicate this information to content editors on all target sites in order to let them decide to translate content again or to delete that version. A way to address this scenario could be to create a custom boolean column for *Page* content type which will be used by content editors on Variation’s source site to mark new versions as “corrections” when necessary. Then, you could create a custom workflow and deploy it on target sites only. This workflow needs to checks every changed publishing page to check the custom column’s value: if value is “True” workflow could simply delete the new version created by Variation’s process on target site.

**Is Versioning Always Required?** In some scenarios target sites may not require approval process to let content go live on a specific site. In that’s scenario a local site admin can disable content approval for publishing pages. Doing that, all pages will “go live” immediately on that target site once Variation’s job is done.

## Search

Enterprise Search in Microsoft Office SharePoint Server 2007 is a Microsoft Office SharePoint Server 2007 shared service that provides extensive and extensible content gathering, indexing, and querying. This service supports full-text searching using Structured Query Language (SQL-based) query syntax, and provides new keyword syntax to support keyword searches. Enterprise Search uses the same underlying Search service as Search in Microsoft Windows SharePoint Services.

On Microsoft Office SharePoint Server 2007 there are a lot of enhancements and improvements on Search. This paper will focus only on that features in Search who are relevant for multilingual content. For more information on Microsoft Office SharePoint Server 2007 Enterprise Search architecture see

*Enterprise Search Architecture****:***

<http://msdn2.microsoft.com/en-us/library/ms570748.aspx>

### Search’ Language Features Overview

Querying and Indexing content are different process provided by search. Each process occurs in a different time and uses resources in different ways:

**Content Crawling:** the index engine uses a pipe of shared memory to request that the Filter Daemon begin filtering the content source. For the crawl process to succeed, the content source must have an associated protocol handler that can read its protocol. The Filter Daemon invokes the appropriate protocol handler for the content source based on the start address provided by the index engine. The Filter Daemon uses protocol handlers and IFilters to extract and filter individual items from the content source. Appropriate IFilters for each document are applied, and the Filter Daemon passes the extracted text and metadata to the index engine through the pipe. At this point in the content crawling process, the index engine saves document properties to a property store separate from the content index. The property store consists of a table of properties and their values. Properties in this store can be retrieved and sorted. In addition, simple queries against properties are supported by the store. Each row in the table corresponds to a separate document in the full-text index. The actual text of a content item is stored in the content index, so it can be used for content queries. The property store also maintains and enforces document-level security that is gathered when a document is crawled. Now, the index engine uses word breakers and stemmers to further process the text and properties picked up during the crawl. The word breaker component is used to break the text into words and phrases. The stemming component is used to generate inflected forms of a given word. The index engine also removes noise words and creates an inverted index for full-text searching.

**Search Query Execution:** when a search query is executed, the query engine passes the query through a language-specific word breaker. If there is no word breaker for the query language, the neutral word breaker is used, which does whitespace-style word breaking, which means that the word breaking occurs where there are whitespaces in the words and phrases. After word breaking, the resulting words are passed through a stemmer to generate language-specific inflected forms of a given word. The use of word breaker and stemmer in both the crawling and query processes enhances the effectiveness of search because more relevant alternatives to a user's query phrasing are generated. When the query engine executes a property value query, the index is checked first to get a list of possible matches. The properties for the matching documents are loaded from the property store, and the properties in the query are checked again to ensure that there was a match. The result of the query is a list of all matching results, ordered according to their relevance to the query words. If the user does not have permission to a matching document, the query engine filters that document out of the list that is returned.

The following diagram provides a detailed view of search’s language features involved at both index and query time:

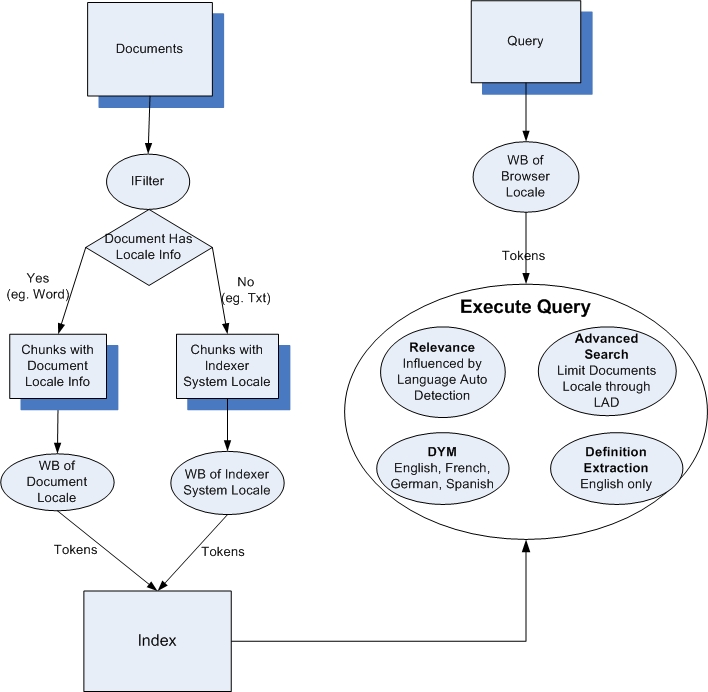


Figure 2: Search Language Features Involvement Flow

Following are the language specific features provided and used by Search Service:

### Word Breakers

A word breaker is a component used by the query and index engines to break compound words and phrases into individual words or tokens. If there is no word breaker for a specific identified language, the neutral word breaker is used, which does whitespace-style word breaking, which means that the word breaking occurs where there are whitespaces in the words and phrases. At indexing time, if there is locale information from the document (example: Word document contains locale information per text run), index engine will try to use the WB for the locale. If the document does not contain locale information, the indexer’s locale is used instead. At query time, the locale of the browser from which query is executed is used to word break queries. Additional information about language availability for word breaker feature is available in *Appendix B:* *Search Language Considerations* at the end of this paper.

### Stemming

Stemming is a feature of Word Breaker component used only by Query engine that determines where the word boundaries are in the stream of characters in the query. A stemmer extracts the root form of a given word. For example, "running," "ran," and "runner" are variants of the word "run." In some languages, a stemmer expands the root form of a word to alternate forms. Stemming is turned off by default for most languages. Stemmers are available for all languages which have morphological expansion. See Appendix B for additional information on Stemming. Additional information about language availability for Stemmer feature is available in *Appendix B:* *Search Language Considerations* at the end of this paper

### Noise Words Dictionary

Noise words are words that do not add value to a query, such as "and," "the," and "a." The indexing engine filters them out to save index space and increase performance. Noise word files are customizable language-specific text files. These files are a simple list of words, one per line. If a noise word file is changed, you must perform a full update of the index to incorporate the changes.

### Custom Dictionary

The custom dictionary file contains values that the search server must include at index and query times. Custom dictionary lists are customizable language-specific text files. These files are used by Search on both Index and query process to identify exceptions to noise word dictionaries. A word like “AT&T”, for example, will never be indexed by default because the word breaker breaks it in single noise words. To avoid this, user can put ‘AT&T’ word in a Custom Dictionary file: as result this word will be threat as an exception by word breaker and it will be indexed and queried. These files are a simple list of words, one per line. If the custom dictionary file is changed, you must perform a full update of the index to incorporate the changes. By default, no custom dictionary file is installed in the SharePoint Portal Server setup procedure.

### Thesaurus

There is a configurable thesaurus file for each language that search supports. Using the thesaurus, you can specify synonyms for words and also automatically replace words in a query with other words that you specify. The thesaurus used will always be in the language of the query, not necessarily the server locale. Additionally, a neutral thesaurus (Tseneu.xml) is applied to every query along with the one associated with the query language when a language without a registered thesaurus is detected.

### Language Auto Detection

Language Auto Detection is the feature that gives best guess on locale of a text chunk based on Unicode range and other language patterns. It’s used basically for relevance calculation by index engine and in queries from Advanced Search web part where you are able to specify which locales can be in a search result.

### Did You Mean

Did You Mean feature is used by query engine to catch possible spelling errors and to provide suggestions for queries. Did You Mean feature basically builds suggestions using the following three components:

1. Query log: information tracked in the query log includes the query terms being used, when search results were returned for search queries and pages that were viewed from search results. This search usage data is beneficial in understanding how people are using search and what information they are seeking. You can use this data to help determine how to improve the search experience for users
2. Dictionary lexicon: a dictionary of most used lexicons provided at installation time
3. Custom lexicon collected from most frequent words in the corpus, built at query time by query engine based on indexed information.

Did You Mean suggestions is available only for the following languages: English, French, German and Spanish.

### Definition Extraction

Definition Extraction feature finds candidate definitions and identifies acronyms and their expansions through examining the grammatical structure of sentences indexed (like NASA, radar, modem and so on). It is available for English language only.

### Diacritics

A diacritical mark or diacritic, sometimes called an accent mark, is a mark added to a letter to alter a word's pronunciation (i.e.. vowel marks) or to distinguish between similar words. A diacritical mark can appear above or below the letter to which it is added, or in some other position; however, note that not all such marks are diacritical. For example, in English, the title (dot) on the letters ‘i’ and ‘j’ is not a diacritical mark, but rather part of the letter itself. Further, a mark may be diacritical in one language, but not in another; for example, in Catalan, Portuguese and Spanish, u and ü are considered the same letter, while in German, Estonian, Hungarian, Turkish and Azeri they are considered to be separate letters. Main usage of a diacritic is to change the phonetic meaning of the letter, but the term is also used in a more general sense of changing the meaning of the letter or even the whole word. Microsoft Office SharePoint Server 2007 supports diacritics. This feature is off by default. Administrators can change this setting through Object Model or stsadm tool. The stsadm command is the following:

**stsadm -o osearchdiacriticsensitive -ssp <ssp name> [-setstatus <True|False>] [-noreset] [-force]**

Additional information about language availability for Diacritics feature is available in *Appendix B:* *Search Language Considerations* at the end of this paper.

### Search Web Parts

Search Service now provides several options that can be useful to scope results and to let users have a multilingual search experience. Following paragraphs describe descriptions about those options and how to use them in multilingual solutions.

#### Windows SharePoint Services V3

There are basically two different search boxes: one for Windows SharePoint Services V3 and one for Microsoft Office SharePoint Server 2007. Depending if Microsoft Office SharePoint Server 2007 is installed on the farm and if Microsoft Office SharePoint Server 2007 Search’s feature is enabled on that site collection (see ***Working with Features*** *at* [*http://msdn2.microsoft.com/en-us/library/ms460318.aspx*](http://msdn2.microsoft.com/en-us/library/ms460318.aspx)for additional information around SharePoint features) user can have a different experience and, for that reasons, customization can be different depending on the environment. Search results in Windows SharePoint Services V3 are always provided by system pages that are at site level. For this reason, there are no customizations to provide since search result’s page will be always in the language of the site. On the other hand, there’s no way to reduce the scope of the search result to get only the content of a single sub-site because, on Windows SharePoint Services V3, search scope covers, now, the entire site collection. For this reason, users will always get information from content on all sites in the site collection. A way to reduce the scope of search results in Windows SharePoint Services V3 is to provide strictly security settings to reduce user’s scope action to a single sub-site. Since search results are trimmed on security, user will see only content on sites where they can access.

#### Microsoft Office SharePoint Server 2007

In Microsoft Office SharePoint Server 2007, by default, all search queries are managed by a special sub-site in the site collection called “Search Center”. Depending on the template you are using in creating a site collection, there are two different Search Center:

* *Rich Search Center* who is provided, by default, by Collaboration Portal template and, since it uses Publishing Feature, can also be used in creating Variation Hierarchy
* *Light Search Center* who is provided by Publishing portal template and doesn’t use Publishing Feature.

Those sites provide search result pages and additional search features like advanced search and, on Rich Search Center only, people search and capability to extend search experience by using additional pages “tabbed” by default using tabs. By default, a single search center is created per site collection. That means the following:

* Search results are always provided using a single language: the language of the search center sub-site.
* All search queries provided from Variation’ target labels use that search center for showing results.

Default behaviors can be changed providing custom settings on Search web parts as following:

**Search Scopes:** Search scopes can be defined at Shared Services Provider level and a site collection level. Scopes defined on Shared Services Provider are *always* *global scopes*. That means that they will be, by default, available in all site collections on all web applications connected to that Shared Services Provider. Site collection administrators, on the other hand, have an option to enable custom scopes that will be available through sites in that site collection. These scopes are completely configurable and can be used by both Search Web Part and Advanced Search Web Part to trim search results. In a multilingual scenario, site collection administrator could create different custom scopes to identify content on each Variation label. Doing this, users would be able to scope a search query to identify only content that is in a specific Variation site.

**Search Box Web Part:** this web part is, by default, located on the following locations:

* *All master pages enabled by Publishing feature*. In these pages, Search Box Web Part has two important characteristics:
  + it is provided as server control
  + It is “surrounded” by another server control called *Delegate Control*. Delegate control is a control that is placed on pages that makes the child control pluggable and replaceable. Additional information on Delegate Control server control can be found here: **Delegate Controls:** <http://msdn2.microsoft.com/en-us/library/ms478826.aspx>
* *Default.aspx* page on search center
* *Results.aspx* page on search center
* *Advanced.aspx* page on search center
* *People.aspx* page on search center (only available on Rich Search Center template)
* *Pepopleresults.aspx* page on search center (only available on Rich Search Center template)

The following sub-set of properties can be useful to provide a multilingual search experience by, for example, limiting the scopes of the search or by redirecting users to a result and advanced search pages located on a search center sub-site under a Variation target site. These properties can be configured on both master pages (using web editor tool like Microsoft Office SharePoint Designer 2007) and on other pages using web UI interfaces:

|  |  |  |
| --- | --- | --- |
| Property | UI Property | Description |
| **DropDownModeEx** | **Dropdown Mode** | This property determines whether to show the scope dropdown or not. If it is selected then what kind of scopes are shown and what is the default selection. More on the details of each possible value are provided below. |
| **AdvancedSearchPageURL** | **Advanced Search Page URL** | This property determines the url for the advanced search page. If no url is specified then the advanced search link is not provided. |
| **SearchResultPageURL** | **Target Search Results Page URL** | This property determines the url of page that will provide search queries results. |
| **UseSiteDefaults** | **Use Site Level Defaults** | This property determines if search box needs to rely on default search settings for behaviors like scopes, and search pages. |
| **ScopeDisplayGroupName** | **Scope Display Group** | This property determines from which display group are authored scopes sourced from. This is a useful property to have few authored scopes to show as options in your search box. By default this property is configured to show global scopes defined at Shared Services Provider level. |

DropDownModeEx property supports the following values:

* *HideScopeDD:*  Do not show scopes dropdown
* *ShowDD:* Show the drop down with both Shared Services Provider defined scopes and site collection defined scopes who match the scopes’ display group defined in *ScopeDeisplayGroupName* property.
* *ShowDD\_DefaultURL:* Same behavior as ShowDD option with providing default selection on what provided in QueryString.
* *ShowDD\_DefaultContextual:* Same behavior as *ShowDD* option with providing default selection on the first scope in the list which is identified as the most relevant contextual scope.
* *ShowDD\_NoContextual:* Shows the drop down with only Shared Services Provider defined scopes.
* *ShowDD\_NoContextual\_DefaultURL:* Same behavior as *ShowDD\_NoContextual* option with providing default selection on what provided in QueryString.

**Advanced Search Box Web Part:** this web part is located on *Advanced.aspx* pages as a web part and, like Search Box web part provides many properties that can be configured using web UI or a web editor tool like Microsoft Office SharePoint Designer 2007. The following sub-set of properties can be specifically useful to provide a multilingual search experience by, for example, limiting the scopes of the search or by redirecting users to a result and advanced search pages located on a search center sub-site under a Variation target site:

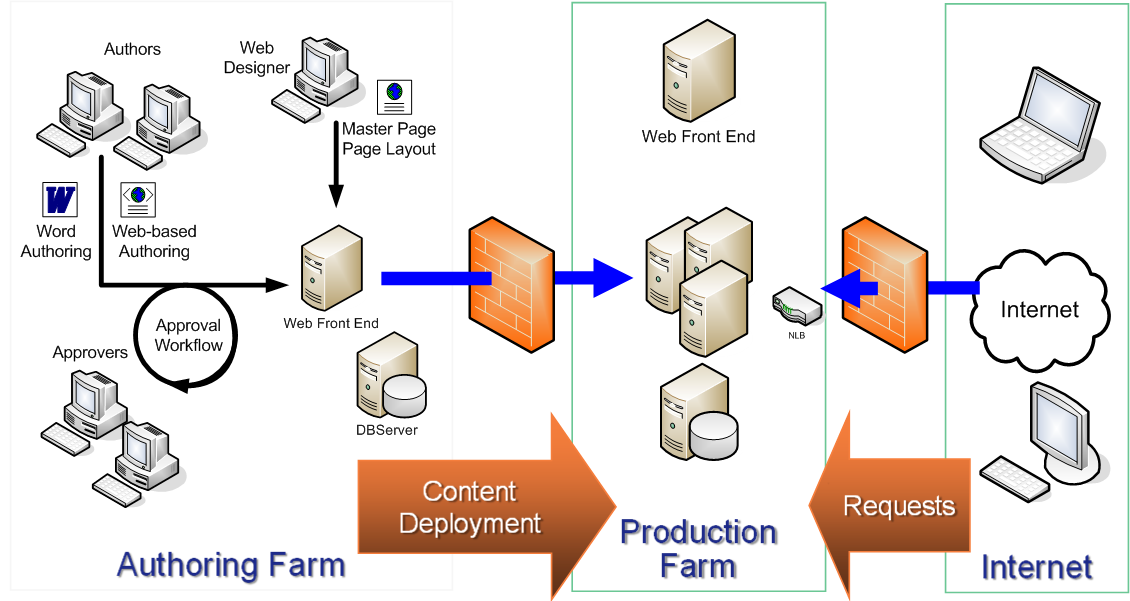
|  |  |  |
| --- | --- | --- |
| Property | UI Property | Description |
| **SearchResultPageURL** | **Target Search Results Page URL** | This property determines the url of page that will provide search queries results. |
| **ShowScopes** | **Show the Scope Picker** | This property determines display scope selection on the advance search box web part. If selected it display by default all scopes that are in the *AdvancedSearch* display group |
| **ShowLanguages** | **Show the Languages Picker** | This property enable language search trimming behavior provided by Language Auto Detection feature to search only documents where content matches specific languages. |

Additional information on Search web parts can be found here: **Customizing the Enterprise Search User Interface** <http://msdn2.microsoft.com/en-us/library/ms546434.aspx>

## Content Deployment

Content Deployment feature in Microsoft Office SharePoint Server 2007 enable site collection content deployment across different site collections than can live on a same farm or in different farms. Content deployment feature is modeled around the idea that a single source site collection deploys content to one or more destination site collections. The destinations therefore contain all the same content that the source site collection contains, with the same structure and organization.

Figure 3: Content Deployment Sample Scenario



### Key Concepts

**Paths and Jobs:** a Content Deployment path defines the relationship between a source and a destination site collection. The source and destination can be in the same server farm or in different server farms. A site collection can have many outgoing paths. Each path has one or more jobs that define the schedule and scope (entire site collection vs. a branch of sub-sites) of individual deployment operations. A job is associated with a path, and it determines exactly which sites in the source site collection will be deployed and on what schedule. You can have many different jobs for a given path, each running on different schedules and deploying specific sections of your site.

**Deployment**: default, deployment only deploys the changes since the last successful deployment, which saves bandwidth and time. And if there aren’t any changes, the deployment completes without redoing any unnecessary work. Dependencies on sites (like custom page layouts) are also managed: Content Deployment feature automatically picks up the dependent page layout and packages it up along with the page itself – even if the dependent resources aren’t in the same site.

**“Quick Deployment”:** Quick Deployment is a special Content Deployment job that is automatically created for every path in any site collection with “Publishing feature” feature enabled. This job, once enabled in a path, wakes up every 15 minutes by default and checks for content that should be deployed quickly. If a content editor who has rights needs to deploy a page quickly, he has an option on the page editing toolbar called “Quick Deploy”. This marks that page to be deployed quickly and Quick Deploy job will deploy it next time it wakes up.

**Security Implications:** Content Deployment jobs work use HTTP or HTTPS protocols to keep content in sync between site collections. Source and destination may have different security contexts; for example, the source might be on an intranet and give permissions to users in the entire company, while the destination may use forms authentication to provide read access to the site. Moreover, source and destination can be in different farms with different topologies. Content Deployment feature doesn’t mind about security context: in configuring paths, user can specify which Windows credentials to use and connect to destination farm and, since Content Deployment feature relies on site collection content in Content Database, destination farm can have a different topology respect of source farm.

### Content Deployment Job’s Process:

Every time a Content Deployment Job runs, the following five steps are always executed:

1. **Check changes log to understanding changes:** once a job successfully completes a schedule, as last step, it updates a log on the source to track status of content on the target. This status is checked by each job at the beginning to understand which content need to be upgraded on target server
2. **Create a new package on the source server:** this package is created based on information in changes log and deployed, temporarily, on file system in source server.
3. **Send the package to the target server:** as described before, this communication uses HTTP or HTTPS protocols (it depends on setting in Content Deployment feature).
4. **Apply changes on target server:** once a package arrives to target server, the Content Deployment job expands it temporarily on a local folder and starts to apply changes. It’s important to understand that this is not a transactional operation: if something happens while changes are applied (a local failure like, for instance, an exception or a power outage) all changes applied successfully will not be rolled back. Next time the same job runs, it will try to re-apply the same content aligning the target to the source
5. **Update Change Log on Source Server:** that’s the last steps that a job performs updating Content status of target site collection on source.

### Constraints and Rules

In configuring the Content Deployment feature is important to keep in mind the following constraints and rules:

**Content Deployment Feature is not Transactional:** As mentioned before, Content Deployment jobs are not transactional and, also, they can only run through all steps:

* + 1. If errors or un-expected behavior happens in executing step number 2, depending of the entity of the error, package can be created partially or can’t be created at all. It’s important to understand that job will terminate only if the package is not created: if the package is created partially, job will always proceed to the following step.
    2. If errors or un-expected behavior happens in executing step number 3, since the package doesn’t reach the target entirely the job ends without applying anything to target.

**Always Ensure Communication:** Communication between source and target (especially in scenarios where they live in different farms) is always important. While a Content Deployment job is running, there is a continuous communication between those Servers. That’s required by Content Deployment feature in understanding job’s status. If, at the end of each step, the target is not able to communicate to source the source will report that the job is in a special state; this means that communication is broken and source is unable to check job’s status on target server. Job status can be interrupted or can still running and succeeded. If after 10 minutes (this is a default that can be changed using Object Model) connection is still broken, the job will be marked as failed; no matter what happened on target server.

Depending on requirements, the Content Deployment feature can surely be used in conjunction with the Variation Feature to enable Geo-deployed multilingual solutions or to migrate existing multilingual CMS 2002 solutions to Microsoft Office SharePoint Server 2007. In these scenarios is important to consider the following:

* For new multilingual implementations the suggested way to implement a geo-deployment solution is to have Content Deployment jobs that deploy the entire site collection vs. deploying only sites for specific languages. That’s because some Variation functionalities, like picker control, require the entire site collection to work properly. Using this approach, it’s possible to have read-only instances of the same site collection geo-distributed across boundaries.
* For multilingual solution migrated from CMS 2002 suggested way to implement full migration is the following:
  + Run migration tools from CMS 2002 to Microsoft Office SharePoint Server 2007. This will create a new site collection in Microsoft Office SharePoint Server 2007
  + On the same site collection set up the Variation feature to creating one Variation’s label for each sub-channel
    - You might need to create a custom template here to have the same experience that you had on CMS 2002.
  + Copy (using Microsoft Office SharePoint Server 2007’s site manager console) the content from the channel identified as “source” to the Variation’s source; this will spawn the content through the hierarchy (after an approval). Then you will have to copy localized content (manually or creating some scripts / tools) from migrates sites into the Variation target sites.

## Custom Web Parts

The following code shows how to create a web part that loads string resources from satellite assemblies. This is the default behavior provided by Windows SharePoint Services V3 and Microsoft Office SharePoint Server 2007 out of the box web parts. Using this technique you can create a single web part that will be used on different sites with different language template packs. To provide localized strings on the web part you will simple create additional satellite assemblies that contain resource strings that will be loaded at run-time by the web part.

This web part provides also an example on how to interact with Variation’s process to fix some internal properties or add additional business logic once a web part is copied to pages on Variation’ target sites by Variation process. The IWebPartVariationUpdate interface provides a method called Update() who is invoked by Variation’s job during page distribution’s process from source site to target sites. Implementing this method, users can interact with Variation’s process if necessary.

Example below relies on the following files:

* *HelloWordMUI.cs:* this file contains the code and logic of the web part
* *MyFirstMUIWebPart.Strings.EN.txt:* contains string resources of web part for English language. This strings will be loaded dynamically by web part when deployed on an English site.
* *MyFirstMUIWebPart.Strings.FR.txt:* contains string resources of web part for French language. This strings will be loaded dynamically by web part when deployed on an English site.

Additional information on how to create satellite assemblies and how to load resource at run-time using ResourceManager class can be found here:

*Resources in Applications****:***

<http://msdn.microsoft.com/library/en-us/cpguide/html/cpconcreatingusingresources.asp>

*Retrieving Resources in Satellite Assemblies*

<http://msdn.microsoft.com/library/en-us/cpguide/html/cpconlocalizingresources.asp>

*ResourceManager Class*

<http://msdn2.microsoft.com/en-us/library/system.resources.resourcemanager.aspx>

Additional information in writing custom web parts in Windows SharePoint Services V3 and Microsoft Office SharePoint Server 2007 can be found on Windows SharePoint Services V3 SDK at the following link:

*Windows SharePoint Services V3 SDK*

<http://msdn2.microsoft.com/en-us/library/ms441339.aspx>

Additional information on PublishingPage class and IWebPartVariationUpdate interface can be found on Microsoft Office SharePoint Server 2007 SDK at the following link:

*Microsoft Office SharePoint Server 2007 SDK*

<http://msdn2.microsoft.com/en-us/library/ms550992.aspx>

HelloWordMUI.cs file

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Text;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Microsoft.SharePoint;

using Microsoft.SharePoint.WebPartPages;

using Microsoft.SharePoint.Utilities;

using System.Xml.Serialization;

using System.Web.UI.HtmlControls;

using Microsoft.SharePoint.Publishing;

using System.Threading;

using System.Reflection;

using System.Resources;

namespace MyFirstMUIWebPart

{

[ToolboxData("<{0}:First MUI WebPart runat=server></{0}:First MUI WebPart>")]

[XmlRoot(Namespace = "MyFirstMUIWebPart")]

//--------------------------------------------------------------------

// File: HelloWrodMUI.cs

//

// Purpose: A sample Web Part that demonstrates how to get language resources from satellite assemblies

// at run time and how to interact with Variation process

//--------------------------------------------------------------------

public class HelloWordMUI : Microsoft.SharePoint.WebPartPages.WebPart, IWebPartVariationUpdate

{

private string variationText = string.Empty;

private string variationTime = string.Empty;

protected override void RenderWebPart(HtmlTextWriter output)

{

// Loads satellite assemblies based on CurrentUICulture of current thread

// Call ResourceManager to get string access to satellite assembly

Assembly resourceAssembly = Assembly.Load("MyFirstMUIWebPart.resources, Version=1.0.0.0, Culture=" + Thread.CurrentThread.CurrentUICulture + ", PublicKeyToken=6fe7a553e3500dfe");

ResourceManager resourceManager = new ResourceManager("MyFirstMUIWebPart.Strings", resourceAssembly);

output.Write("<TABLE width=\"100%\" class=\"ms-main\" CELLPADDING=0 CELLSPACING=0 BORDER=0>");

//Get string information from satellite assembly using ResourceManager

output.Write(resourceManager.GetString("HelloWordMUI\_Intro"));

//Get the current SharePoint contentxt to display it on the web part

string tmpContext = string.Empty;

try

{

tmpContext= SPContext.GetContext(HttpContext.Current).Web.Url;

}

catch( Exception ex)

{

tmpContext = string.Format(resourceManager.GetString("HelloWordMUI\_Error"), ex.StackTrace);

}

output.Write(string.Format(resourceManager.GetString("HelloWordMUI\_Current"), tmpContext));

if (string.Compare( VariationMessage, string.Empty)!= 0)

{

output.Write(string.Format(resourceManager.GetString("HelloWordMUI\_VarHome"), VariationMessage));

output.Write(string.Format(resourceManager.GetString("HelloWordMUI\_SpawnTime"), LastVariationSpawn));

}

output.Write("</TABLE>");

}

//Define two hidden properties that are used as propertybag by web part

//This properties are populated by Variation's copy process once happened

//Contains the last time when Variation's process occurred

[Browsable(false), WebPartStorage(Storage.Shared)]

public string LastVariationSpawn

{

get

{

return variationTime;

}

set

{

variationTime = value;

}

}

//Contains the URL of the Variation's Home site.

[Browsable(false),WebPartStorage(Storage.Shared)]

public string VariationMessage

{

get

{

return variationText;

}

set

{

variationText = value;

}

}

#region IWebPartVariationUpdate Members

//This method is invoked by Variation's process when this web part is copied on a page on a target site

//This web part uses this method to set up custom properties that will be showed on pages on target sites

//if the web part has been copied by Variation's process.

public void Update(PublishingWeb ownerWeb)

{

// Loads satellite assemblies based on CurrentUICulture of current thread

// Call ResourceManager to get string access to satellite assembly

Assembly resourceAssembly = Assembly.Load("MyFirstMUIWebPart.resources, Version=1.0.0.0, Culture=" + Thread.CurrentThread.CurrentUICulture + ", PublicKeyToken=6fe7a553e3500dfe");

ResourceManager resourceManager = new ResourceManager("MyFirstMUIWebPart.Strings", resourceAssembly);

try

{

string txtTmpUrl = ownerWeb.ParentPublishingWeb.Url;

this.VariationMessage = txtTmpUrl;

this.variationTime = Convert.ToString(DateTime.Now);

}

catch (Exception ex)

{

this.VariationMessage = string.Format(resourceManager.GetString("HelloWordMUI\_Error"), ex.StackTrace);

}

}

#endregion

}

}

MyFirstMUIWebPart.Strings.EN.txt file

HelloWordMUI\_Current=<TR><TD>Current Location: {0} Url.</TD></TR>

HelloWordMUI\_Error=Bad Error:{0}!

HelloWordMUI\_Intro=<TR><TD>MUI Sample Web Part Provides information about where it is.</TD></TR>

HelloWordMUI\_SpawnTime=<TR><TD>Last Variation time: {0}.</TD></TR>

HelloWordMUI\_Title=Hello Word in English

HelloWordMUI\_VarHome=<TR><TD>Variation Home is: {0}.</TD></TR>

MyFirstMUIWebPart.Strings.FR.txt file

HelloWordMUI\_Current=<TR><TD>endroit courant: {0} Url.</TD></TR>

HelloWordMUI\_Error=Erreur Critique:{0}!

HelloWordMUI\_Intro=<TR><TD>C'est un example de Web Part qui fourn renseignement on ou elle se trouve.</TD></TR>

HelloWordMUI\_SpawnTime=<TR><TD>bout temp: {0}.</TD></TR>

HelloWordMUI\_Title=Boujour a Tout le Monde

HelloWordMUI\_VarHome=<TR><TD>La maison de la variation se trouve ici: {0}.</TD></TR>

# Appendix A: Languages

The following table shows release languages for both Windows SharePoint Services V3 and Microsoft Office SharePoint Server 2007 will be released included language packs.

|  |  |  |
| --- | --- | --- |
| **Language** | Language Release | Language Template Pack |
| Arabic | **X** | X |
| Bulgarian |  | X |
| Catalan (LIP) |  | X |
| Chinese (Simplified) | **X** | X |
| Chinese (Traditional) | **X** | X |
| Croatian |  | X |
| Czech | **X** | X |
| Danish | **X** | X |
| Dutch | **X** | X |
| English | **X** | X |
| Estonian |  | X |
| French | **X** | X |
| German | **X** | X |
| Greek | **X** | X |
| Hebrew | **X** | X |
| Hindi\* |  | X |
| Hungarian | **X** | X |
| Italian | **X** | X |
| Japanese | **X** | X |
| Korean | **X** | X |
| Latvian |  | X |
| Lithuanian |  | X |
| Norwegian | **X** | X |
| Polish | **X** | X |
| Portuguese (Brazil) | **X** | X |
| Portuguese (Portugal) | **X** | X |
| Romanian |  | X |
| Russian | **X** | X |
| Serbian (Latin) |  | X |
| Slovak |  | X |
| Slovenian |  | X |
| Spanish | **X** | X |
| Swedish | **X** | X |
| Thai\* | **X** | X |
| Turkish | **X** | X |
| Ukrainian |  | X |

# Appendix B: Search Language Considerations

The following are some consideration about search service features and components

## Word Breakers

The following languages support native Word Breaker and Stemmers in Microsoft Office SharePoint Server 2007. Other languages are threat by Neutral Word Breaker

|  |
| --- |
| **Language** |
| Arabic |
| Bulgarian |
| Catalan (LIP) |
| Chinese (Simplified) |
| Chinese (Traditional) |
| Croatian |
| Czech |
| Danish |
| Dutch |
| English |
| Estonian |
| French |
| German |
| Greek |
| Hebrew |
| Hindi\* |
| Hungarian |
| Italian |
| Japanese |
| Korean |
| Latvian |
| Lithuanian |
| Norwegian |
| Polish |
| Portuguese (Brazil) |
| Portuguese (Portugal) |
| Romanian |
| Russian |
| Serbian (Latin) |
| Slovak |
| Slovenian |
| Spanish |
| Swedish |
| Thai\* |
| Turkish |

## Diacritics Support

Diacritics are supported by Search engine in following languages

|  |
| --- |
| **Language** |
| Albanian (Albania)  Azeri (Latin)  Basque  Belarussian  Bengali (Bangladesh)  Bengali (India)  Bulgarian  Catalan  Croatian (Croatia)  Croatian (Bosnia and Herzegovina)  Czech  Danish  Estonian  Faeroese  Filipino  Finnish  French (Belgium)  French (Cameroon)  French (Canada)  French (Côte d'Ivoire)  French (France)  French (French Congo (DRC))  French (Haiti)  French (Luxembourg)  French (Mali)  French (Monaco)  French (Morocco)  French (North Africa)  French (Reunion)  French (Senegal)  French (Switzerland)  French (West Indies)  Gaelic (Irish)  Gaelic (Scots)  Galician  Gallegan  German (Austria)  German (Germany)  German (Liechtenstein)  German (Luxembourg)  German (Switzerland)  Greenlandic  Hawai'ian (United States)  Hindi  Hungarian  Icelandic  Inari Sámi (Finland)  Inuktitut  Italian (Italy)  Italian (Switzerland)  Kazakh (Cyrillic)  Kazakh (Latin)  Konkani  Kyrgyz (Latin)  Latin  Latvian  Lettish  Lithuanian  Lule Sámi (Norway)  Lule Sámi (Sweden)  Luxembourghish  Macedonian (FYRO)  Maltese (Malta)  Māori (New Zealand)  Mapudungun  Marathi  Northern Sámi (Finland)  Northern Sámi (Norway)  Northern Sámi (Sweden)  Norwegian (Bokmål)  Norwegian (Nynorsk)  Oriya  Papiamentu  Polish  Portuguese (Brazil)  Portuguese (Portugal)  Punjabi  Quechua (Bolivia)  Quechua (Peru)  Quechua (Ecuador)  Rhaeto-Romanic  Romanian (Moldava)  Romanian (Romania)  Russian  Russian (Moldava)  Serbian (Latin, Bosnia and Herzegovina)  Serbian (Latin, Serbia and Montenegro)  Skolt Sámi (Finland)  Slovak  Slovenian  Sothern Sámi (Sweden)  Southern Sámi (Norway)  Spanish (Argentina)  Spanish (Bolivia)  Spanish (Chile)  Spanish (Colombia)  Spanish (Costa Rica)  Spanish (Dominican Republic)  Spanish (Ecuador)  Spanish (El Salvador)  Spanish (Guatemala)  Spanish (Honduras)  Spanish (Latin America)  Spanish (Mexico)  Spanish (Nicaragua)  Spanish (Panama)  Spanish (Paraguay)  Spanish (Peru)  Spanish (Puerto Rico)  Spanish (Spain) (International; Sort)  Spanish (Spain) (Traditional Sort)  Spanish (United States)  Spanish (Uruguay)  Spanish (Venezuela)  Swedish (Finland)  Swedish (Sweden)  Tagalog  Turkish  Turkmen (Cyrillic)  Turkmen (Latin)  Ukrainian  Uzbek (Cyrillic)  Uzbek (Latin)  Vietnamese  Welsh (United Kingdom)  Yakut |

## Language Auto Detection Support

Language Auto Detection, included language selection in Advanced Search Page, is provided by default for following languages

|  |
| --- |
| **Language** |
| Arabic  Bengali  Bulgarian  Catalan  Chinese  Croatian/Serbian  Czech  Danish  Dutch  English  Finnish  French  German  Greek  Gujarati  Hebrew  Hindi  Hungarian  Icelandic  Indonesian  Italian  Japanese  Kannada  Korean  Latvian  Lithuanian  Malay  Malayalam  Marathi  Norwegian  Polish,  Portuguese  Punjabi  Romanian  Russian  Slovak  Slovenian  Spanish  Swedish  Tamil  Telugu  Thai  Turkish  Ukrainian  Urdu  Vietnamese |