



SIMIAN systems

Multilingual Implementation Guide

Sitellite Content Management System

Introduction

Sitellite 5 features all-new multilingual capabilities that enable it to serve your content in any number of languages. These capabilities include tools and workflow integration for translators, including a side-by-side reference interface for editing translated text, as well as the ability to easily integrate these new features into existing Sitellite websites and modules upon upgrading to Sitellite 5.

Integrating the new multilingual features into a newly upgraded or an all-new Sitellite website can be done in a matter of minutes, then the real work of translating all of the content can begin.

This guide will show you how to implement these new multilingual features into a new or existing Sitellite website and the considerations and capabilities of this new functionality.

Considerations

Storage and Performance Considerations

Sitellite breaks down the translation of a website, its content and its modules into two types of translation formats: file and database storage.

Content is considered any of the various content types in Sitellite including web pages, news stories, events, sidebars, products as well as custom content types (also known as “collections”). These are stored in the database, specifically in the “sitellite_translation” and “sitellite_translation_sv” tables. These actually represent a content type unto themselves, which includes its own versioning capabilities.

Each piece of content has an associated translation in this collection for each language/locale combination. The translations each themselves adopt the same permissions (team ownership and access level) as the original content, except for the status which is specific to the translation. This allows a translation to be in “draft” status when the original has already been approved.

Everything else is considered to be a “string” of text, which is simply stored in a file with all of the text “strings” for that language/locale combination for each module. This file can more quickly be interpreted by Sitellite than individual database queries for each piece of text, thereby ensuring optimal performance, especially since there can be dozens or even hundreds of individual “strings” for a given application.

Since these files are PHP-formatted, it also has the side benefit that they can be cached into the server's RAM memory by a PHP bytecode cache such as the APC cache extension, which will significantly improve their speed on APC-enabled Sitellite sites.

Languages, Locales and Character Sets

Each language has an optional locale or specific region to which it belongs, which allows for regional dialects of a single language. Languages also have an associated character set, which tells the visitor's computer how to render a given language.

For many languages, the "UTF-8" character set works well, however others require a less generic character set. Sitellite allows you to specify the character set that it uses for each language in the language menu of the translations tool.

Language Fallbacks

Fallback languages and locale-specific languages allow for greater specificity and better matching of languages to visitor preferences. For example, if you offer your website in French but also specific French translations for different locales (e.g., French-Canadian), then you can specify the generic French as a fallback to the locale-specific translations, and depending on the visitor's browser preferences it will choose the correct one based on their locale, falling back to the generic language where needed.

This allows you to translate the majority of the copy in generic form and simply customize the appropriate text where needed for each locale as well, so you don't have to re-translate each individual piece of text for each locale, except where that local translation differs from the generic form.

If no translation of a text string or piece of content is available, the default language will be shown.

Translator Interface and Workflow

Translators are users who have the role "translator", although additional roles can be granted permission to access the translation tools as well. For example, a "master" user always has access to all of the available tools.

Web View

Sitellite provides translators with a simple way of using the Web View to browse your website for copy requiring translating. When a page or sidebar or news item (or any other type of content) requires translation, they can simply click on the pencil icon and it will bring them to a translation language selection and then to the translator interface to translate that piece of content.

Control Panel

In the Control Panel, translators will find the translation tools under the Tools > Translations menu option.

This interface allows translators to manage all aspects of a site's languages. The main screen is simply a list of content that can be translated. It is broken down by Content Type, Language, and Translation Status. Translation statuses include:

- **Not Translated** – Content that has no translation for that particular item and language selection.
- **Incomplete** – Content that has a translation that has not been marked “approved”.
- **Complete** – Content that has been translated and approved for the live site.
- **Expired** – Content that is still “approved” but has been subsequently updated in the original language.

To translate a given document, simply select the language to translate to in the left-most column and the translation form will open for that document. The translation tools do not limit a translator to a particular language so that one translator may handle multiple languages.

On the translation form, the left column contains the form fields to fill out and the right column contains the reference content from the original language. For wysiwyg fields such as the web page body, the reference content will automatically scroll to keep itself in sync with the translated text, to help improve the productivity of translators. This feature can be disabled/re-enabled through the “Synchronize scrolling” checkbox at the bottom of the reference content.

The last field in the form is the “Translation Status”, which must be set to “Approved” for the content to begin appearing on the live website. If it is left in “Draft” it can be saved and completed later.

The other areas of the translation tools are the “Languages” screen where the available languages are managed, the “Global Templates” screen where individual text “strings” for each language can be translated, and

the “Applications” screen where text “strings” for each module can be translated.

In the Global Templates and Applications screens, there is an “Update Index” link which will extract all of the text strings for a given module and save them to the translation files. This must be run once for each module before it can be translated.

Please ensure with your system administrator that the server’s folder permissions are correct, or the Update Index feature will not be able to run. The permissions required are on the global inc/lang folder as well as the lang subfolder for each module in the inc/app folder and they must allow the web server (Apache and PHP) to write to those folders and any files in them. This commonly means 0777 but that setting can vary on different servers.

Workflow

To enable the workflow for translators, go to Control Panel > Admin > Workflow Services and ensure the workflow service “Translator Workflow Notifier” is checked.

Translators are automatically notified by email and in their Satellite inbox of any new documents added or modified in the original language. They can follow the link in the message and it will bring them directly to the translation page for the new or modified content.

Visitor Language Configuration

There are two main techniques for choosing the right language to show to a visitor, HTTP negotiation and using cookies.

HTTP Negotiation

The default is HTTP negotiation, which compares the available languages on the website to those in the visitor’s browser preferences to determine the best language to show for them. This is a popular technique used by large websites like Google, however since it is completely automatic on the website it requires that they have their web browser correctly configured with their language preferences, which some users especially those on shared computers, may not.

Cookies

The other option is to use cookies and provide the visitor with a language selection box. The steps to enabling users to choose their language preference themselves and track it through a cookie are as follows:

1. As a master user, go to Control Panel > Admin > Site Settings and change the Language “Negotiate” setting to “Cookie”.
2. Next, go to Control Panel > Tools > Translations and click on the “Languages” link in the top left. Create the languages you will be offering on your website and ensure the correct one is marked as the default. Note that you can always add new languages from this screen at any time.
3. In the Web View, create a sidebar box where you would like the language selection to appear. Give the sidebar an ID of “language” and a title like “Choose a language”.
4. In the body of the sidebar, click on the Box Chooser icon. Select “Utilities” then “Language Selector” in the box chooser. This will insert a dynamic box into the sidebar body that will automatically generate the language selection for you.
5. Under the “State” tab, set the box to be public and approved and click “Save” at the bottom of the form.

You should now have a fully working multilingual website using cookies and offering visitors their preference of language. The next step is simply to get started translating your site!

Translating Custom Components

Sitellite makes it easy for custom module developers to make their modules multilingual as well. Here are the aspects to consider in developing or converting your module to be multilingual compatible:

Inline Text in Templates

In global (.tpl) template file, to mark a piece of text as translatable through the translation tool, simply wrap it in the `<xt:intl>` tag like this:

```
<xt:intl>This text is translatable.</xt:intl>
```

In module-specific templates (.spt) files, to mark a piece of text as translatable, wrap it in an `{intl}` tag like this:

```
{intl This text is translatable.}
```

Inline Text in PHP

In your PHP code if you have text that needs to be translated, you can pass it to the `intl_get()` function like this:

```
echo intl_get ('This text is translatable.');
```

Alternately, you can embed dynamic elements into the text like this:

```
echo intl_getf (
    'Today is the %d day of the month.',
    strftime ('%e')
);
```

This uses PHP's `sprintf()` function to embed the text into the translated string.

Form Fields

Sitellite will automatically read and make translatable any form field names specified in your form settings.php files.

Custom Content Types

Content types defined in the `inc/app/cms/conf/collections` folder are automatically translatable by Sitellite. However, you still need to retrieve the correct translation for each document when serving them through your custom modules. There are two strategies for doing so, one of which is mostly automatic and the other requires additional code inserted where you retrieve content from the database.

Using the `saf.Database.Generic` package to abstract the retrieval of content from a particular database table or set of tables, you can substantially minimize the amount of code you need to write for common select, insert and update type queries. `Generic` also understands Sitellite's internal permissions on collections that have them simply by setting the `usePermissions` property to `true`. The same goes for `Generic`'s awareness of Sitellite's multilingual features. Simply set the `multilingual` property to `true` and `Generic` will take care of the rest. For example:

In the news module's `NewsStory` class (`inc/app/news/lib/Story.php`) the constructor method is defined as:

```
function NewsStory () {
    parent::Generic ('sitellite_news', 'id');
    $this->usePermissions = true;
    $this->multilingual = true;
}
```

With this code, we can now include the `NewsStory` class into our own code and select news stories without worrying about the details of Sitellite's permissions or multilingual features at all. For example:

```
loader_import ('news.Story');
$news = new NewsStory ($parameters['story_id']);
echo template_simple ('show_story.spt', $news->makeObj ());
```

This code will automatically retrieve the story with the id from the `story_id` parameter passed to the script, with the correct language and permissions (although we skipped any error checking on the permissions!). Generic takes care of the rest for you.

The other way involves a bit more code, but Sitellite makes this manageable too, via the `multilingual.Translation` package. If we didn't specify that Generic should use its own automatic multilingual awareness, we could still do this:

```
// import the translation package
loader_import ('multilingual.Translation');

// specify the collection and language
$t = new Translation ('sitellite_news', intl_lang ());

// admins can see drafts since they may want to edit them
if (session_admin ()) {
    $translated = $t->get ($parameters['story_id']);
} else {
    $translated = $t->get ($parameters['story_id'], true);
}

// check if we've got something
if ($translated) {
    foreach ($translated->data as $key => $value) {
        echo $key . ' = ' . $value . '<br />';
    }
}
```

As you can see, the PHP code required to translate individual content items is still very straight-forward, but Generic certainly makes it easier. For more information about Generic and its many time-saving features, check out www.sitellite.org.

Single-Value Tables

For database tables that don't contain enough information to warrant a full content type (even one with `visible=no` so they're not shown in the Control Panel > Content menu), such as lists of categories, tags, topics, sections, etc., Sitellite provides a facility for application developers to enable these to be easily translated as well.

In your application's conf folder, simply create a file named `translate.ini.php` which contains a list of database tables and the field you want to make translatable in them, for example:

```
siteevent_category = name
siteevent_audience = name
```

Simply rebuild the index and the name field for each of those two tables is now translatable in the Translation tool along with the regular text strings for your application.

To access them programmatically, you can use the same `intl_get()` function as before and for `.spt` templates you would instead use:

```
{filter intl_get}{name}{end filter}
```

With each of these techniques, Sitellite makes it very easy for developers to create multilingual applications or to enable multilingual capabilities in existing Sitelite-based applications.