C:\DOCUME~1\hakhtar\LOCALS~1\Temp\GrvTempzwugxigansrz4a2gheui1\C\vamosa_logo_final_280809.tif

Sitecore 6 Connector Guide

Using the Vamosa Sitecore 6 Connector

<http://www.vamosa.com/>

1. Document Control
2. Change Record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Author | Version | | Change Reference | Date |
| A Mancevice | | v1.0 |  | 1/5/2009 |
|  | |  |  |  |

1. Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Company | Job Title and/or Project Role | Date Reviewed |
|  |  |  |  |
|  |  |  |  |

1. Approvers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Company | Job Title and/or Project Role | Date Reviewed |
|  |  |  |  |
|  |  |  |  |

1. Terms & Conditions

Copyright © 2010 Vamosa Ltd

All rights reserved, no part of this document may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Vamosa.

This document is for the use of the recipient(s) only. None of its content may be disclosed to third parties without the prior written permission of Vamosa. Such disclosure will only be made on the basis that the contents are not passed to persons other than those specified in the consent. In addition, neither the whole nor any part of this document, nor any reference thereto may be included in any document, circular or statement without the prior written approval by Vamosa of the form and context in which it will appear.

1. Table of Contents

[1 Introduction 4](#_Toc250451297)

[1.1 Document Purpose 4](#_Toc250451298)

[1.2 The Connector Package 4](#_Toc250451299)

[2 Set Up 4](#_Toc250451300)

[2.1 Deploying the Connector 4](#_Toc250451301)

[2.1.1 Client Side 4](#_Toc250451302)

[2.1.2 Vamosa Side 4](#_Toc250451303)

[3 Loading Content 4](#_Toc250451304)

[3.1 Pages 4](#_Toc250451305)

[3.2 Assets 5](#_Toc250451306)

[4 Additional Resources 5](#_Toc250451307)

# Introduction

## Document Purpose

This document will explain how to deploy and use the connector files for migrating into the Sitecore 6 Content Management System. These procedures must be completed prior to the development phase of a migration and are required to ensure a path from source to target environments is established and therefore facilitate a successful end to end migration. As a consequence, the execution of these steps is a key indicator to the project state and readiness to migrate.

## The Connector Package

The supplied package contains:

* The C# code behind the web service
* An asmx file that will represent the web service as a web page
* The java library for using the connector in VCM
* The python script to call the web service from VCM
* A helper script to view item fields

# Set Up

## Deploying the Connector

### Client Side

Deploying the connector in the client’s environment involves gaining access to the machine where the Sitecore instance resides. After establishing this connection, navigate to the target instance in the filesystem (ex. C:\Inetpub\wwwroot\<client\_site\_name>\WebSite) and copy the file “VamosaService.asmx” to the directory and place “VamosaService.dll” in the “/bin” sub-directory. The web service should now be accessible by retrieving the page “http://<client\_site\_address>/VamosaService.asmx”.

### Vamosa Side

Before VCM can be used to migrate content the Sitecore-specific Java library must be deployed in the C:\vamosa\server\default\lib directory and the server restarted. The script “Sitecore Connector.py” should be used in a loading task pipeline.

# Loading Content

## Pages

Loading pages with data can be accomplished in one pass but to successfully translate links[[1]](#footnote-1) the load must be executed in two steps (placeholder and content). To create a placeholder the content descriptor requires three metadata values given by “Target.Template Path” (the path to the page template for a given content descriptor), “Target.Path” (the directory where the content will be placed), and “Target.Name” (the name of the Page).

To load content the descriptor must be transformed into an XML document with the root “<item>” and subsequent “<field>” children. Each field should have a “name” attribute which corresponds to the name of the field in the Sitecore GUI. Further attributes may be used to indicate different types of fields but is not necessary.

The connector is such that pages may be loaded in any order with non-loaded parents being created as folders (and then updated when processed by the connector). Executing a placeholder load on a previously created page will have no effect other than returning the previously created page’s Item ID (ie, no duplicates).

## Assets

Assets are loaded using the same script as pages but only require “Target.Path” and “Target.Name” fields. Note that the “Target.Path” value is assumed to begin under the Media Library. Like pages, assets’ directory structure is created on the fly by the Sitecore API.

# Additional Resources

An additional helper script has been provided to aide in the proper transformation of content field values. All field values are represented as strings in Sitecore but some take on properties that are not immediately apparent (Dates, for instance, require a specific format not explained through the GUI alone). This script is intended to be run across a single object whose fields have been manually populated through the GUI and returns the item’s field values as a string for reference in building the transformation step. Manually populating the “Date” field of the page with the path “/content/Home/some\_Item” with today’s date (1/5/2010 10:47, for instance) followed by an execution of the Get Item Fields script should display the Date field’s value as 20090105T104700. This technique can be used with any field value (banner images, dates, etc).

1. A note on link translation: Sitecore uses a special link engine such that links to pages should be translated to “~/link.aspx?\_id=<Item ID>&amp;\_z=z” and links to assets to “~/media/<Item ID>.ashx” [↑](#footnote-ref-1)