

The logo for Portal Technology, featuring the word "Portal" in white text on a dark blue square background, followed by the word "Technology" in dark blue text on a white rectangular background.

SiteGenerator Help and User Guide

Portal Technology Systems Limited

United House

North Road

London N7 9DP

Tel: +44 207 6099411

Fax: +44 207 9002864

Email: info@portaltech.net

Version: 1.3

Date: 11 January 2002

© Portal Technology Systems Ltd, 2002

<http://www.portaltech.net>

The information contained herein is confidential. Any duplication, retransmission, dissemination or other use of this information is strictly prohibited.

© Portal Technology Systems Limited 2002

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, reduced to any electronic medium or machine-readable form, or be recorded or transmitted in any way or by any means without the written permission of the copyright holder.

This documentation is provided “as is” and may include technical inaccuracies or typographical errors. Portal Technology Systems Limited may make changes or improvements to this publication at any time and without notice. Any such changes or improvements will be contained in the new edition of this publication.

Portal Technology Systems Limited limits any liability to the maximum extent permitted by law.

Index

Version History.....	5
References.....	5
Introduction	6
What is SiteGenerator	7
Installing and Starting SiteGenerator	8
Installation	8
Running SiteGenerator as a GUI program	8
Running SiteGenerator as a command line program or script.....	8
What SiteGenerator needs	11
A Template File	11
Template contents for 'Insert Entire Page Text' Option.....	11
Template contents for 'Insert Path/Filename' Option	12
Page Fragments.....	13
Links and relative paths	13
Using SiteGenerator	13
The User interface.....	15
Buttons	15
Browse.....	15
Build:.....	16
Clear All	16
Exit.....	16

Menu Items.....	17
File > Save Settings.....	17
File > Exit.....	17
View > Log.....	17
Settings > Page Generation Method	17
Settings > File Separator	17
Settings > Look and Feel.....	18
Help > About.....	18
Logs and Properties	19
Properties file	19
Log file.....	19

Version History

Date	Author	Comment	Version
04-Jun-2001	PDD	Document creation.	1.0
30-Jun-2001	AMW	Applied document template and reviewed.	1.1
04-Jul-2001	PDD	Added changes following review. Added section: Running SiteGenerator as a command line program	1.2
11-Jan-02	PDD	Added extra command line options to SiteGenerator following SiteGenerator build-1-0-3	1.3

References

[1] <http://www.java.sun.com>

[2] <http://jakarta.apache.org/ant/>

Introduction

This document contains user documentation relating to the SiteGenerator product produced by Portal Technology Systems Limited. It explains how the SiteGenerator can quickly aid the rebranding or redesigning of a website.

What is SiteGenerator

Websites often try to produce a consistent look and layout across several pages. Although the content displayed on each page changes, things such as a logo, navigation links, other graphics and background images may be constant for each page throughout the site.

When such a site requires one of these consistent elements to be changed it can prove time-consuming if each page has been separately constructed.

One solution to this problem is to produce a template that contains all of the core elements of the site. There would be a place in the template where the content for a page can be inserted.

The solution then involves automating a way to generate pages based on this template, from a series of page fragments containing the content for each page. Subsequent changes to the template can then be made, the pages automatically regenerated, and the site is instantly updated.

SiteGenerator provides a way of doing this. It generates pages, based on the template, which include the fragment pages. There are two ways that SiteGenerator can generate these new pages:

- Insert the entire page fragment text
- Insert the path and filename of the page fragment

The second method requires that the website can serve pages that allow some form of `include` statement, such as in a JSP.

The `include` statement in JSP takes the form:

```
<%@ include file="filename" %>
```

When a page containing an `include` statement is served, the text contained in the specified *filename* is included at that point of the page.

When the option to insert filenames is selected, SiteGenerator inserts the relative path to the file followed by the filename into the pages it creates.

Installing and Starting SiteGenerator

Installation

SiteGenerator is distributed as an executable .jar file. This means that it requires that the Java 2 Runtime Environment, or a full version of the Java 2 SDK installed on the system [1].

The file SiteGenerator.jar is all that is needed. No installation is necessary, other than saving this file in the location that you want.

Running SiteGenerator as a GUI program

SiteGenerator runs by default as a GUI program, requiring that a native GUI operating system is available, for example Windows, for it to operate as a GUI.

To start it from within GUI environment (e.g. Windows), double click on the SiteGenerator.jar file. SiteGenerator should begin executing. This may not work if files with .jar extensions are not registered to run the java environment.

To start it from a command line change to the directory in which SiteGenerator resides and type the following command:

```
java -jar SiteGenerator.jar
```

SiteGenerator should begin executing.

Running SiteGenerator as a command line program or script

SiteGenerator can be executed from the command line, with the parameters required for a build being entered as command line parameters. To run the SiteGenerator from the command line enter the following:

```
java -cp SiteGenerator.jar pts.sitegenerator.SiteGenerator <params>
```

where <params> specifies the parameters that will be used in the build. The parameters expected by SiteGenerator are:

<source> <dest> <template> <tag> [<file separator>] [<page generation method>]
 [<input file type>] [<output file type>]

where:

Parameter	Description
<source>	the source directory
<dest>	the destination directory
<template>	the template file (including the full path)
<tag>	the template tag to replace
<file separator>	optional parameter specifying the file separator used in the generated pages. Valid values are 'unix' or 'dos'.
<page generation method>	optional parameter specifying the page generation method used to generate pages. Valid values are 'text' or 'filename'.
<input file type>	optional parameter specifying the file type of the source files. If it does not begin with a '.' then one is added.
<output file type>	optional parameter specifying the file type of the output files. If it does not begin with a '.' then one is added.

To include any of the optional parameters it is necessary to specify all of the parameters (including optional ones) that **precede** it in the above list.

For a fuller explanation of these parameters see the following sections.

This enables SiteGenerator to be executed from scripts and allows it to be part of an automated process.

In this way SiteGenerator could be run multiple times from a script, each using a different set of parameters. For example a site may have several different groups of pages, each with a different template.

For example of using SiteGenerator from a script, SiteGenerator could be run from an ANT build.xml file [2] as follows:

```
<target name="generate">
  <!-- SiteGenerate pages -->
  <java classname="pts.sitegenerator.SiteGenerator" fork="on">
    <classpath>
      <pathelement path="${lib}/SiteGenerator.jar"/>
    </classpath>

    <!-- Source Directory -->
    <arg value="${build}/web-app/pages"/>

    <!-- Destination Directory -->
    <arg value="${build}/web-app"/>

    <!-- Template File -->
    <arg value="${build}/web-app/WEB-INF/templates/template.jsp"/>

    <!-- Tag -->
    <arg value="fragment_link"/>

  </java>
</target>
```

What SiteGenerator needs

SiteGenerator builds pages based on a template and a series of page fragments containing the content for each generated page. It does this by finding a user specified tag in a template file and replacing it with either:

- The entire page fragment text
- The relative path and filename of the page fragment

It then saves this generated page and repeats the process for all of the page fragments.

It requires the following things to be set up:

A Template File

A template file contains the elements of the site that will be consistent from page to page. The template should contain a unique piece of text (called a tag, for convenience sake), which will be replaced by SiteGenerator. Exactly what it is replaced with depends on the page generation method selected.

If 'Insert entire page text' has been selected as the page generation method, SiteGenerator will replace the tag with all of the text contained in the fragment.

If 'Insert path/filename' has been selected as the page generation method, SiteGenerator will replace the tag with the relative path from the destination directory to the page fragment, followed by the filename.

Template contents for 'Insert Entire Page Text' Option

At the point where the content will go, insert a unique piece of text (referred to as the tag). Site generator will replace this text with the text contained in the page fragment for each generated page.

```
<!-- EXAMPLE TEMPLATE -->
<!-- To be used with SiteGenerator -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
    "http://www.w3.org/TR/REC-
html40/loose.dtd">
<html>
<head>
  <title>Portal Technology Systems Limited</title>
```

```
<meta http-equiv="Content-Type" content="text/html;">
</head>

<body>

    <!-- A HEADING COULD GO HERE -->

    <!-- NAVIGATION LINKS COULD GO HERE -->

    <fragment_link>

    <!-- LOGO COULD GO HERE -->

</body>
</html>
```

The point at which the page fragment text will be inserted is called the *tag* and in the above example it is `<fragment_link>`.

Template contents for 'Insert Path/Filename' Option

At the point where the content will go, insert an `include` statement which will refer to the page fragment. Because this statement will point to a different fragment for each page of content, SiteGenerator will need to place the filename and path for each page here. It does this by replacing a 'tag' defined by the user.

As an example, using a JSP page, the following could constitute a template:

```
<!-- EXAMPLE TEMPLATE -->
<!-- To be used with SiteGenerator -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
    "http://www.w3.org/TR/REC-
html40/loose.dtd">
<html>
  <head>
    <title>Portal Technology Systems Limited</title>
    <meta http-equiv="Content-Type" content="text/html;">
  </head>

  <body>

    <!-- A HEADING COULD GO HERE -->

    <!-- NAVIGATION LINKS COULD GO HERE -->

    <%@ include file="<fragment_link>"%>

    <!-- LOGO COULD GO HERE -->

  </body>
</html>
```

The point at which the filename containing the page content will be inserted is called the *tag* and in the above example it is `<fragment_link>`.

Page Fragments

Any pages containing content should be grouped together in a directory and should just contain the content that is to be placed in the template generated page. They can have any kind of file extension, but should all have the same one. It is important that they are all in the same directory, not using any subdirectories as SiteGenerator does not recursively search through subdirectories.

Links and relative paths

It should be noted that any links to images, other pages etc. in the page fragments must be relative to the destination directory, not the source directory.

If a page fragment also contains an `include` statement, particular care needs to be taken that the URL specified in it is appropriate for the 'Page generation method' that will be used:

- If Page Generation Method is set to 'Insert Path/Filename only' then the URL needs to be relative to the source directory.
- If Page Generation Method is set to 'Insert entire page text' then the URL needs to be relative to the destination directory.

Using SiteGenerator

When SiteGenerator is run, the user is asked to specify a number of parameters:

- The source directory is the directory that contains all of the page fragments.
- The destination directory is the directory where the generated pages will be placed.
- The template file is the file containing the template on which the generated pages will be based.
- The tag is the text in the template file that will be replaced with the filenames of the page fragments.
- The input file type is the file extension type of the files contained in the source directory.

- The output file type is the file extension type of the files that will be generated in the destination directory.

When these parameters have been entered and the build button is clicked, SiteGenerator generates new pages in the destination directory based on the template and the page fragments in the source directory.

The page generation method used (Path/Filename insert or Entire text insert) depends on the settings selected.

The User interface

The user interface for SiteGenerator is shown below:



The text entry boxes correspond to the parameters referred to in the previous section. The source/destination directory and the template file can be selected either typing the entry in the box, or by clicking on the adjacent browse button and selecting the appropriate directory or file.

An invalid entry will cause SiteGenerator to display an error message.

The entries for input/output file type may optionally start with a '.' SiteGenerator will automatically add one if it is not present.

Buttons

Browse

Each of the browse buttons will open up a file open dialog box. SiteGenerator will place the selected file/directory into the adjacent text field. If an inappropriate selection is made, the user is directed to make another selection:

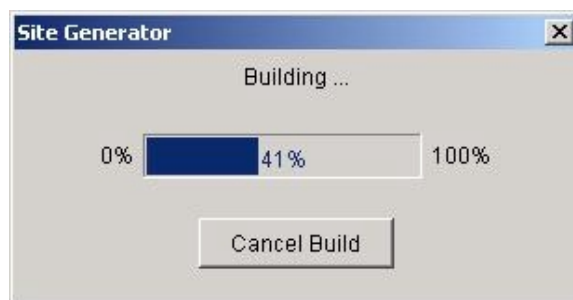
**Build:**

This commences generation of the pages based on the parameters supplied.

Firstly the directories and template file are checked to see that they are valid, and an error message displayed if they are not. Secondly, the template is checked to ensure that it contains the tag and again an error message is displayed if it is not:



If all parameters are correct, then the build commences and a progress dialog is displayed with an option to cancel the build. Once the build is complete, the cancel button changes to an OK button:

**Clear All**

This button clears the text boxes for source/destination directories, template and tag.

Exit

This button shuts down and exits SiteGenerator.

Menu Items

File > Save Settings

This option will save all of the current values for settings and parameters from the text fields into the Properties File.

File > Exit

This option is the same as the Exit button.

View > Log

This will display a copy of the log file in a new window. For information about the log file see the next section. This window must be closed to resume using SiteGenerator.

Settings > Page Generation Method

This setting determines the method that SiteGenerator will use to generate its pages. There are two choices:

- Insert Path/Filename only: SiteGenerator will replace the tag in the template with:
 - the relative path from the destination directory to the source directory followed by:
 - the page fragment filename.
- Insert entire page text: SiteGenerator will replace the tag in the template with:
 - the entire text contained in the page fragment.

Settings > File Separator

This option is only available when the Page Generation Method is set to 'Insert Path/Filename only'.

There are two kinds of file separators in current use: Unix style systems use a '/' and Windows/Dos uses a '\\'.

This option does *not* apply to the entries in the text fields for the source/destination and template. Rather, it specifies the file separator that SiteGenerator should use when building the path and filename to replace the tag in each generated page.

The appropriate separator will be determined by the operating system environment in which the website will be hosted.

Settings > Look and Feel

This enables SiteGenerator to change its appearance to mimic alternative GUI environments. The default setting is System, which uses the current operating system defaults.

Help > About

This displays an about box.

Logs and Properties

SiteGenerator creates two files and places them in the same directory in which SiteGenerator is located.

Properties file

The first file is a properties file called SiteGenerator.properties. This contains a record of the parameters and settings used in the last successful build. When SiteGenerator is started, it looks in this file, loads parameters from it into the text entry fields, and uses the specified settings.

Log file

The other file that SiteGenerator creates is a log file called SiteGenerator.log. This contains logging information recorded since SiteGenerator was last started. It is the file which is displayed by the menu option Settings > View Log.