

# Content Server URL Assemblers including LightWeight Abstract Assembler Item Context Assembler Site Plan Tree Support

Author: Tony Field

Created: July 23, 2009

# Overview

Writing URL assemblers for Content Server can be a tedious task. This library exists for two reasons: to simplify the work required to develop an assembler, and to create a set of multi-purpose, easy-to-use assemblers that can be used in the vast majority of sites without requiring that any code be written.

# Executive Summary

A new abstract assembler (base class) has been written that is much easier to use than the abstract assembler that ships with Content Server. This facilitates development of new assemblers.

A generic assembler, called ItemContextAssembler, is applicable for any site needs to represent a single object placed in a specific context in its URL. This applies to most sites. For example, a web page describing downloads for a specific product in a catalogue has the download page as its item and the product as the context. Alternately, articles on pages could be the items in context.

A helper class that supports ItemContextAssembler when used with sites based on the Site Plan Tree is provided. It includes an infrastructure for converting an asset into a pretty alias using a variety of pluggable strategies. Many strategies are included with this assembler. Examples would be the conversion of the asset ID Page:123123123123 into an alias such as “about-us”. The Site Plan Helper will concatenate aliases into a context and determine which path to use automatically, and all the developer has to do is choose or create a new mapping strategy, which takes a fraction of the time required to develop a new assembler.

# Component Overview

## LightweightAbstractAssembler

The LightweightAbstractAssembler class greatly simplifies the work required to write a URL assembler. Handling property configuration, default value support, URL encoding, decoding, and query string processing, it handles all of the tedious tasks dealing with URL assembly, leaving only the core business logic to be developed by the developers.

Unlike its predecessor, com.fatwire.cs.core.uri.AbstractAssembler, the complexity of utilizing this helper class is gone. There are no callbacks, no complex sub-interfaces. The assemble and disassemble methods are not implemented in this class – it only provides clear helper methods.

## ItemContextAssembler

**Overview**

The ItemContextAssebler is designed to be a general purpose URL assembler that is applicable whenever a URL needs to be constructed by putting an item in a context. It does not support many other features, but the applicability of this approach is very broad.

Examples of single items in contexts are:

* A file in a folder
* An article on a site plan tree Page
* A product in a catalogue
* A press release on a given date

Three key parameters drive this assembler: “item-context”, “item-type”, and “item-alias”. They are concatenated together in a structure that looks like this:

<item-context>/<item-type>/<item-alias>

If this structure is generally suitable, this assembler is applicable.

**Restrictions**

Besides requiring the three parameters described above, the assembler requires that the URL be a Satellite Server URL, designed to serve pages, not images or other blobs. Additionally, this assembler requires that pagename and childpagename match a configured value. If all of these conditions are met, the assembler will attempt to assemble the URL. If all of these conditions are not met, it will delegate URL assembly to the backup assembler, which as of July 24, 2009 is defined as the QueryAssembler. Perhaps in the future the backup assembler can be configurable.

**Usage**

While this assembler can be used unmodified in all of the above situations, a nagging problem exists in that “item-context”, “item-type” and “item-alias” are not standard variables used by Content Server using any standard rendering models. However, they do naturally map to some. All that is needed is a conversion mechanism to map known variables into these three item-\* parameters. This mapping is done through a series of helper classes.

As of July 24, 2009, only the Site Plan Tree Helper exists. However, it is conceivable that others could be added and ship with this package at a later date. These helpers take input parameters and convert them into the item-\* parameters, and then back again. Because this conversion often requires database access, it is not possible for the conversion to occur in the assembler itself. Instead, conversion of site params into ItemContextAssembler params must be done at the time the link is created – usually just before invoking the render:calltemplate tag. In the FSII rendering model, this occurs in Link templates. The reverse lookup must occur before the site params are needed. Often the params are needed right away, which means the conversion back has to happen in a wrapper page or in a layout template (depending on caching and security requirements).

**Multivariate Testing Support**

This assembler has support for integer-based variant parameter, which is appended to the end of the URL using a suffix “/v<int>” (e.g. /v1). This feature transparently allows multivariate testing by allowing this parameter to be passed on a URL (without requiring a query string, which is helpful for working with some popular web analytics packages) and converted into a variable that can be used in templates to differentiate page layouts, views, or content, etc. The variant parameter must be an integer, and it is optional.

**Configuration**

The ItemContextAssembler contains several configuration parameters that address the definition of the required pagename and childpagename, a mapping system for item-type parameters (optional), the item type used by the item-context, and finally, a list of arguments that should always be unpacked from the ugly packedargs parameter.

The default configuration file is the ServletRequest.properties file in cs.war/WEB-INF/classes.

**For more information**

For more information about exactly how to use this assembler, consult the Java API Reference Guide for com.fatwire.developernet.uri.itemcontext.ItemContextAssembler.

## Site Plan Tree Helper

## AliasingStrategies