FATWIRE INTERACT 2010
GLOBAL USER CONFERENCE

Search Engine Optimization Using Content Server

Tony Field
Global Solutions Architect

Agenda

- **Overview of SEO techniques**
  - “Fly Casual”
  - Quality Content
  - Keyword research & placement
  - Avoid redundancy by not repeating yourself
  - Page Structure

- **How FatWire Can Help**
  - Collaborative authoring
  - Asset Modeling
  - Flex Filters & Text Mining
  - Cross-linking, SEO and URLs
  - Template Design Techniques & the GSF
  - Site Map
  - Utilize UGC

- **Discussion**

Part I: An Overview of Selected SEO Topics

Search Engine Optimization is the use of various techniques to improve a web site's ranking and thus attract more visitors

-- Wiktionary

“Fly Casual”

- **Search Engine Optimization = End User Optimization**
  - Make your pages easy for your visitors and search engines will reward you
  - The most compelling pages nearly always rank first
- **Be a good citizen**
  - Don’t bother trying to trick the search engines
  - …they know about you…
  - Google changes its algorithm 500 times per year
- **Know what search engines are trying to do but don’t act like you know how they are doing it**

Quality Content

- **Compelling content always ranks higher**
  - Why do you like the pages you visit every day?
    - It’s about a topic you find interesting
    - You haven’t seen it before
    - Other people with similar interests find it relevant

- **Keep your content fresh and relevant**
  - Fresh content represents your contribution – your value
  - Focus
    - Off-topic rants don’t help your credibility

Keyword Research & Placement

- **Think back to high school**
  - Organize your thoughts into an outline
  - Create a keyword outline
    - Select relevant key words for your content
    - Research their relevance and value
    - If your content is organized, these outlines will coincide
  - Merge your outlines together into a well-crafted, organized article, with keywords sprinkled in appropriate places

Stay On Topic

- Keep the theme of your page consistent & launch new pages for new ideas
- Diluting the theme of your page dilutes its value to your users, and therefore its page rank

I once knew a man who kept talking about things that didn't matter. Boy did that drive me nuts. He was such a character. I'll never forget him. He had brown hair, walked with a limp, and whenever he spoke the strangest thing would happen: His right eye would blink in morse code, spelling out the words He was actually saying

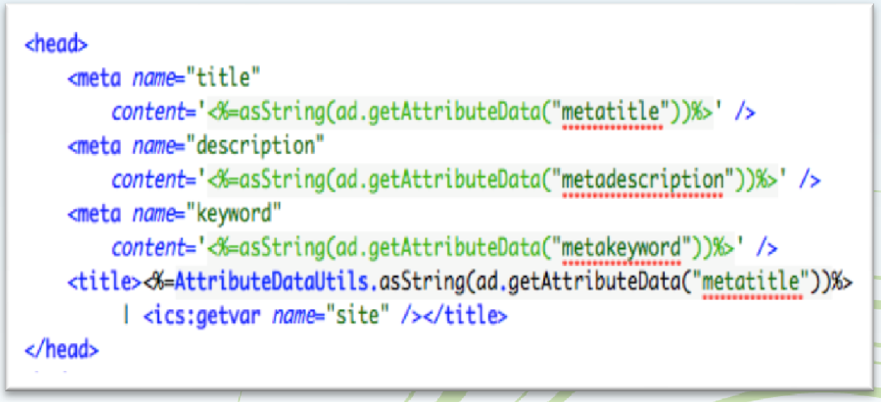
?!?!?

Avoid Dilution and Duplication

- **You will be penalized in the rankings if your content appeared elsewhere and is duplicated without any added value**
  - Even if it’s your own content and both pages are yours (don’t split your own page rank)
- **Avoid having more than one URL for a page**
  - www.fatwire.com vs fatwire.com
  - Printer-friendly URLs
  - Duplicate URLs
  - All of these will dilute your page rank by splitting it among the different URLs

Page Structure

- Craft your markup as thoughtfully as your content
- Use title tags
- Use h1/h2 tags
- Images aren’t read – use alt text
- A word about meta keywords
- Write a good meta description

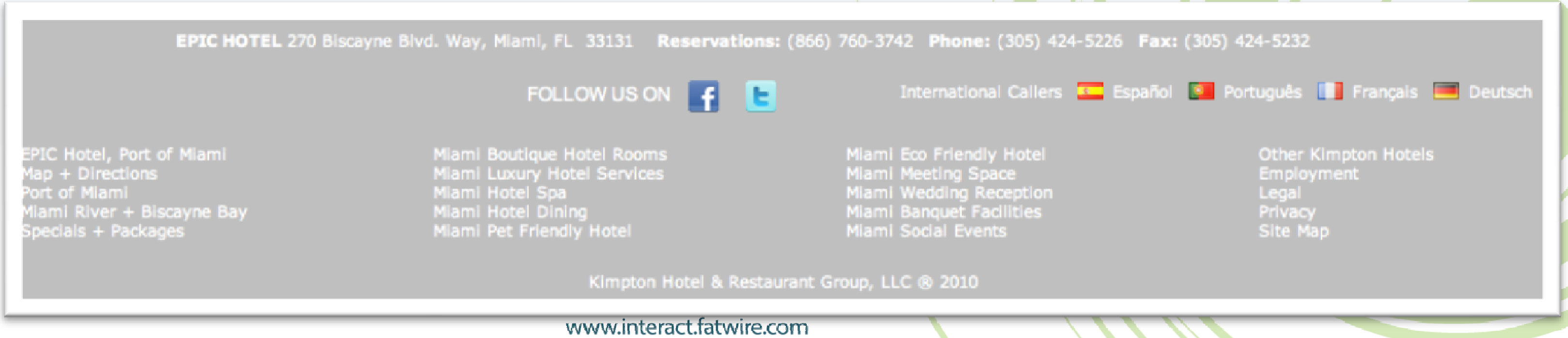


```
<head>
    <meta name="title"
        content='<%=asString(ad.getAttributeData("metatitle"))%>' />
    <meta name="description"
        content='<%=asString(ad.getAttributeData("metadescription"))%>' />
    <meta name="keyword"
        content='<%=asString(ad.getAttributeData("metakeyword"))%>' />
    <title><%=AttributeDataUtils.asString(ad.getAttributeData("metatitle"))%>
        | <ics:getvar name="site" /></title>
</head>
```

Page Structure

- **URLs**
  - Actual URL content
  - Uniqueness
  - URL Shortening Services
- **Links are endorsements**
  - Consider the source!
    - Yourself, a respected site, a link farm…
- **Use a site map page for visitors**
- **Have a sitemap.xml**

- **Show your physical address!**
  - So you can be be accurately placed in local results pages.



EPIC HOTEL 270 Biscayne Blvd. Way, Miami, FL 33131 **Reservations:** (866) 760-3742 **Phone:** (305) 424-5226 **Fax:** (305) 424-5232

FOLLOW US ON

International Callers Español Português Français Deutsch

EPIC Hotel, Port of Miami
Map + Directions
Port of Miami
Miami River + Biscayne Bay
Specials + Packages

Miami Boutique Hotel Rooms
Miami Luxury Hotel Services
Miami Hotel Spa
Miami Hotel Dining
Miami Pet Friendly Hotel

Miami Eco Friendly Hotel
Miami Meeting Space
Miami Wedding Reception
Miami Banquet Facilities
Miami Social Events

Other Kimpton Hotels
Employment
Legal
Privacy
Site Map

Kimpton Hotel & Restaurant Group, LLC ® 2010

Part II

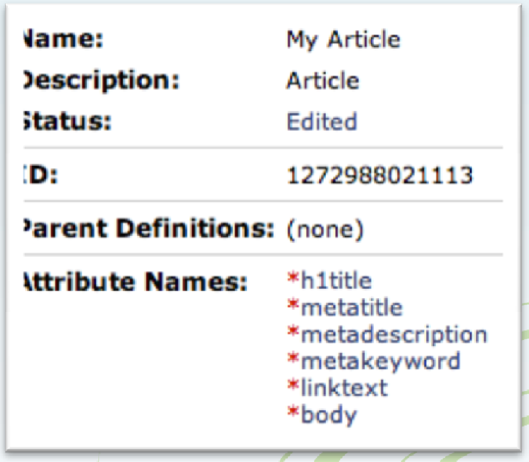
How SEO can be Implemented With FatWire's Software

Write Compelling Content

- Utilize TeamUp to formulate your content and keyword outlines
- Collaborate with others in your organization before writing your copy
- When ready, move content to Content Server

Asset Model

- **Your asset model should capture key data**
  - H1 title
  - Meta desription
  - URL
  - Keywords
  - Link text
  - Tags
- **Define relationships to link to related content**
- **Carefully manage external links as assets so that they can be used strategically**



Name:	My Article
Description:	Article
Status:	Edited
ID:	1272988021113
Parent Definitions:	(none)
Attribute Names:	*h1title *metatitle *metadescription *metakeyword *linktext *body

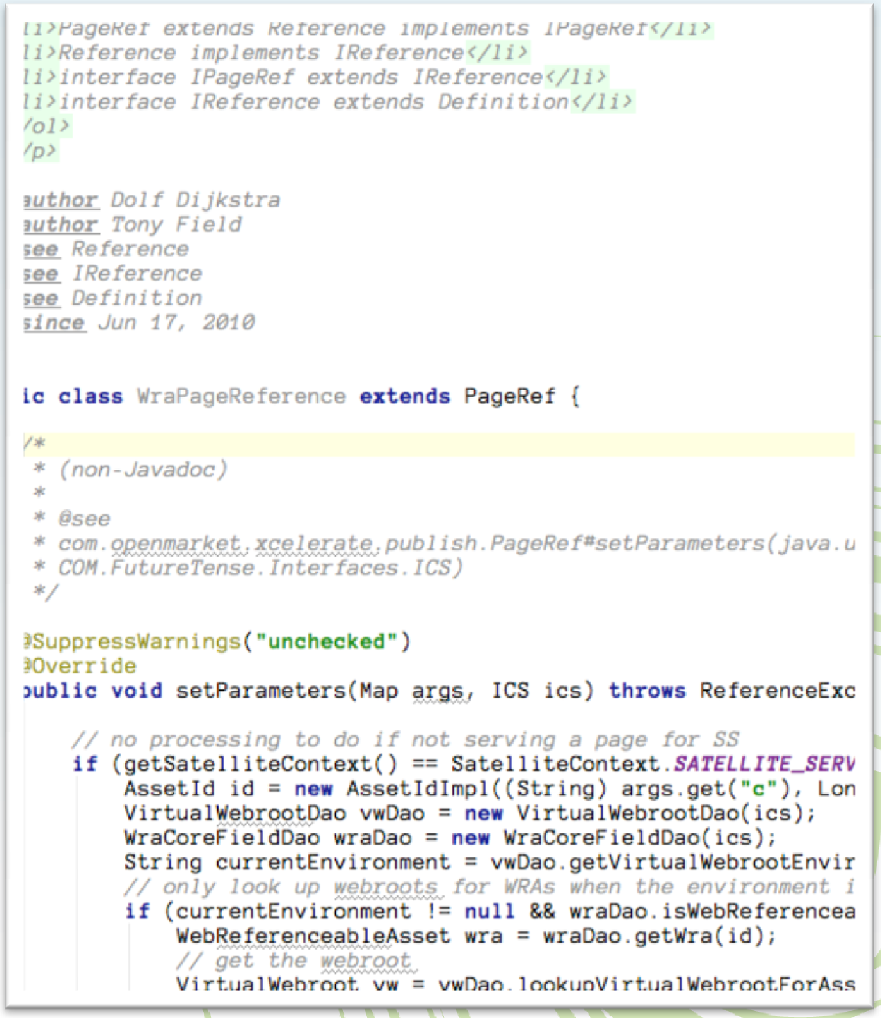
GSF supports all of these out-of-the-box with Web Referenceable Assets, Tagging, and Aliases

Utilize Flex Filters for Automation and Text Mining

- **Users don’t always create their own meta descriptions or identify relevant keywords**
- **A flex filter is called just before a flex asset is saved to the database**
- **Invoke external services to process data and update asset fields**
  - Extract the first 2 sentences for a description
  - Invoke a text mining service to extract keywords
- **Customize the UI to leverage link suggestions**

Link API Improvements

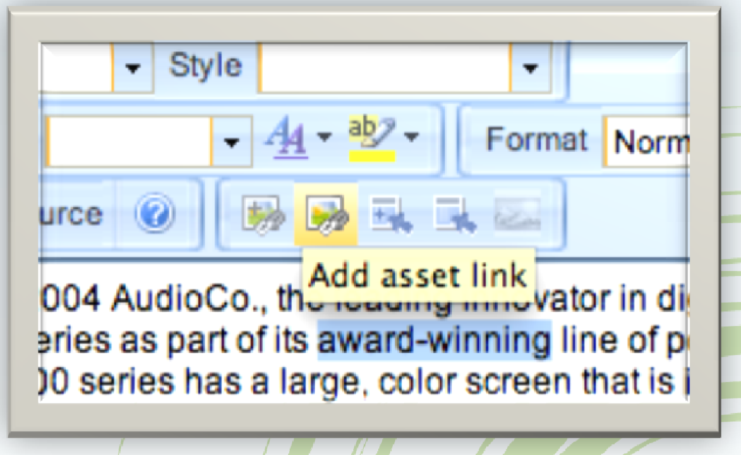
- **In older rendering models, c, cid & p were often required**
    - Users needed to use Link templates to calculate proper links
    - Embedding links in text meant embedding a Link pagelet
    - Linktext had to be passed in
- **Since 7.5.3, we can extend the link calculation mechanism to do this logic in Java**
    - No more Link templates!
    - Enables “highlight and link” in rich text fields



```
li>PageRef extends Reference implements IPageRef</li>
li>Reference implements IReference</li>
li>interface IPageRef extends IReference</li>
li>interface IReference extends Definition</li>
/ol>
/p>

author Dolf Dijkstra
author Tony Field
see Reference
see IReference
see Definition
since Jun 17, 2010


ic class WraPageReference extends PageRef {

/*
 * (non-Javadoc)
 *
 * @see
 * com.openmarket.xcelerate.publish.PageRef#setParameters(java.u
 * COM.FutureTense.Interfaces.ICS)
 */

@SuppressWarnings("unchecked")
@Override
public void setParameters(Map args, ICS ics) throws ReferenceExc

    // no processing to do if not serving a page for SS
    if (getSatelliteContext() == SatelliteContext.SATELLITE_SERV
        AssetId id = new AssetIdImpl((String) args.get("c"), Lon
        VirtualWebrootDao vwDao = new VirtualWebrootDao(ics);
        WraCoreFieldDao wraDao = new WraCoreFieldDao(ics);
        String currentEnvironment = vwDao.getVirtualWebrootEnvir
        // only look up webroots for WRAs when the environment i
        if (currentEnvironment != null && wraDao.isWebReferencea
            WebReferenceableAsset wra = wraDao.getWra(id);
            // get the webroot
            VirtualWebroot vw = vwDao.lookupVirtualWebrootForAss
```

Effective Cross-Linking

- **The introduction of the concept of Web Referenceable Assets in the GSF adds to this**
  - Only one URL per asset
  - Therefore no need to pass “p”
  - *All WRA links are proper, vanity-url-style permalinks*
- **Link generation frustration is history**
- **Editors can link to their own content using keywords they select from their body text, which improves search ranking**



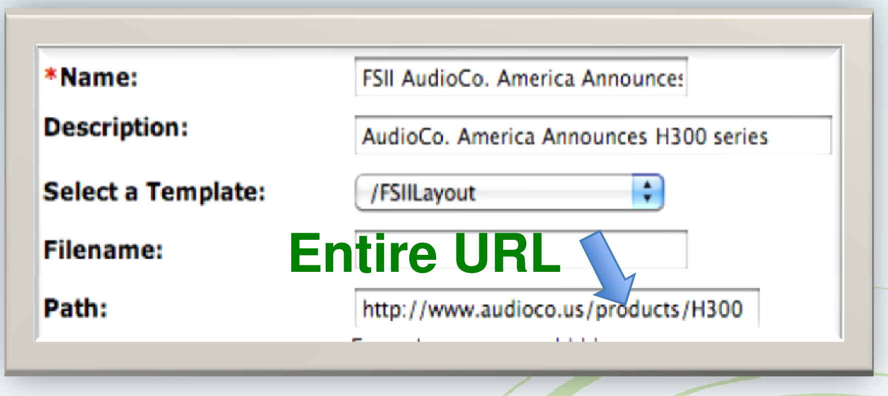
004 AudioCo., the leading innovator in di
eries as part of its award-winning line of p
00 series has a large, color screen that is i

Search Engine-Optimized URLs

- **Fully hand-crafted (vanity) URLs are highly desirable for web marketers**
- **Search engines favor URLs with appropriate keywords in them**
  - Pretty is not essential for search engines, but humans like them
- **URL assemblers introduced this capacity several years ago**
  - Tradeoffs existed
    - Custom code was often required
    - Configuration was tricky
- **This year the GSF introduced the Web Referenceable Asset and associated URL assembly infrastructure**
  - Full vanity URL support is included
  - Complete control is provided simply and transparently to editors

Full Control over URL Format

- The path field now contains the *entire* URL for the WRA as it would be deployed to production
- Smart substitution infrastructure ensures the URL can work on other environments
- Wide open text field?
  - URL must match a pre-registered “Virtual Webroot” prefix
  - Workflow or UI customizations can further restrict editing
  - Auto-suggest URLs based on headline etc.



*Name:	FSII AudioCo. America Announces
Description:	AudioCo. America Announces H300 series
Select a Template:	/FSIILayout
Filename:	
Path:	http://www.audioco.us/products/H300

Entire URL

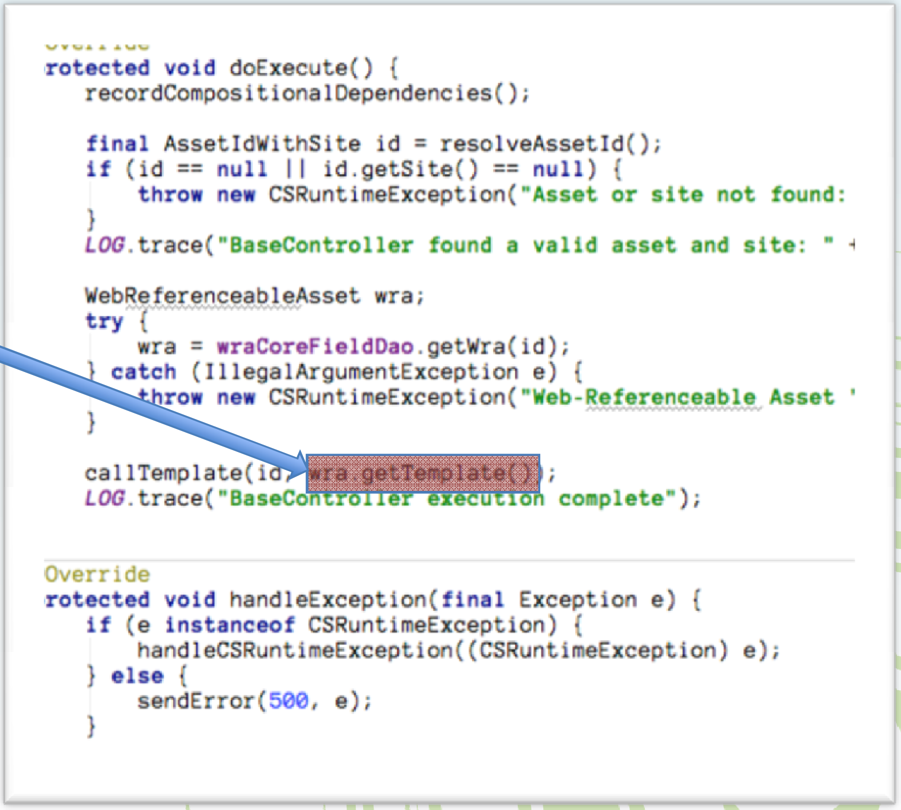
- Customize the path field UI for
  - Auto-suggestion
  - Restriction
  - Full automation

Template Design Techniques

- **Clean template design is difficult when combined with pagelet caching**
  - The FirstSite II rendering model has a Layout template
    - Beyond that, tracing through code to identify which code produced markup is tedious
  - Template layouts that override the standard Layout require special cases that tend to be developer-unfriendly
    - If-then-else blocks
    - Multiple nested layers
  - Maintenance costs of this complex infrastructure outweigh the performance and re-use benefits
- **A simpler, more pragmatic approach was needed!**
  - Use template-dispatching directly from the wrapper (i.e. no Layout!)
  - This is the technique utilized by the GSF

Template Design Techniques & the GSF

- **Rather than avoiding duplicating code at all costs (which are too high), allow some**
- **The WRA specifies its own template & the controller calls it.**
  - Eliminates *Layout* and *Detail dispatcher* layers completely
- **All core layout code moves into the WRA main body template**
  - Everything is visible to the developer in one place
  - Normal caching applies
  - Virtually no tradeoff

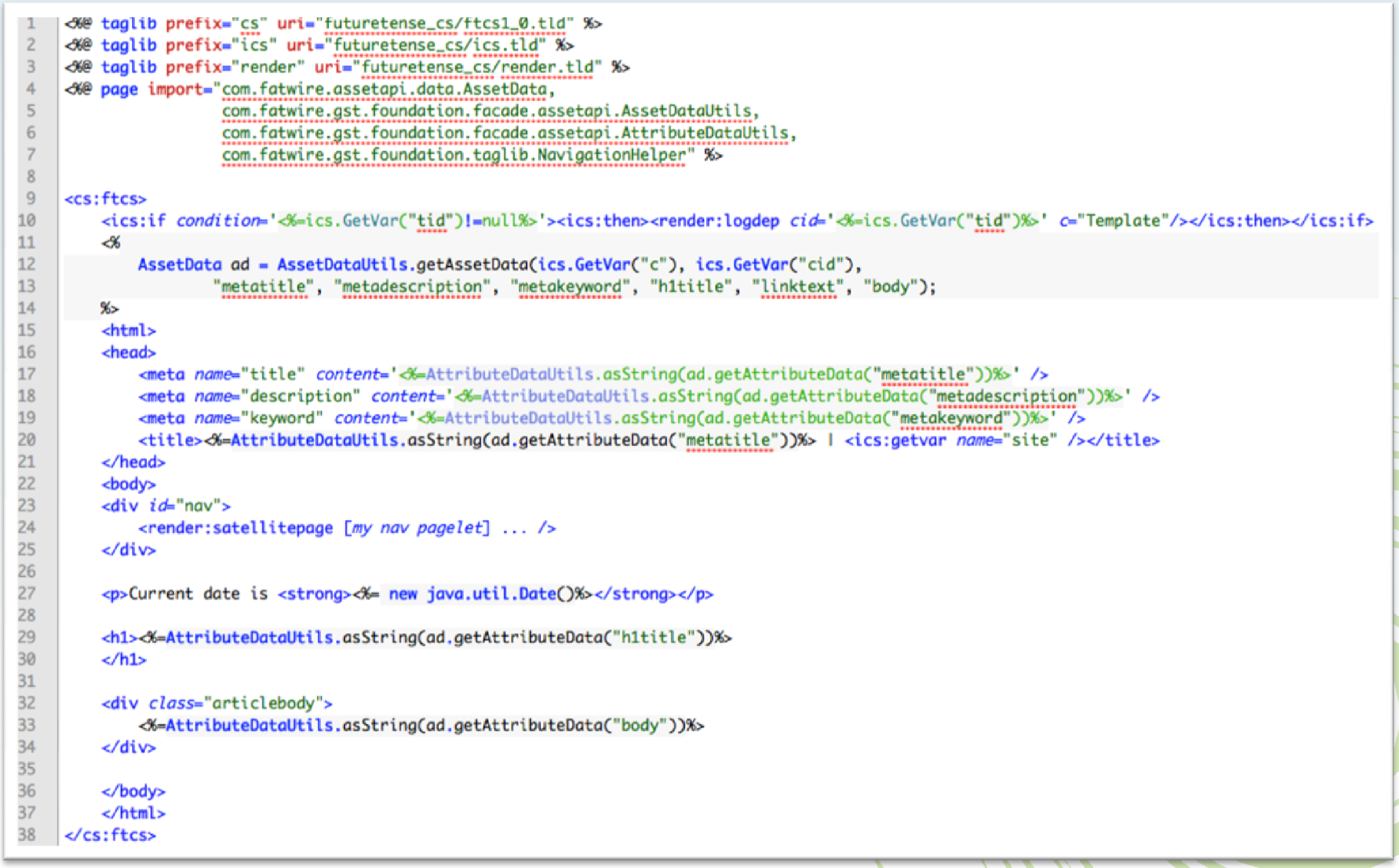


```
Override
rotected void doExecute() {
   recordCompositionalDependencies();

   final AssetIdWithSite id = resolveAssetId();
   if (id == null || id.getSite() == null) {
       throw new CSRuntimeException("Asset or site not found:
   }
   LOG.trace("BaseController found a valid asset and site: " +

   WebReferenceableAsset wra;
   try {
       wra = wraCoreFieldDao.getWra(id);
   } catch (IllegalArgumentException e) {
       throw new CSRuntimeException("Web-Referenceable Asset '
   }

   callTemplate(id, wra.getTemplate());
   LOG.trace("BaseController execution complete");


Override
rotected void handleException(final Exception e) {
   if (e instanceof CSRuntimeException) {
       handleCSRuntimeException((CSRuntimeException) e);
   } else {
       sendError(500, e);
   }
```

Template Design Techniques

- **I know what you're thinking…**
- **"You're telling me I have to duplicate my Layout code!"**
- **Yes. I am.**
  - Weigh the tradeoff:
    - Most sites have 8-20 wireframes
    - Of those, 2-6 are "special"
    - As few as 2 and only as many as 18 end up being duplicated
    - Pagelets are still regularly used for headers, footers, navs, etc.
    - Thus a very small number of lines end up being duplicated, and they are presented very clearly to the developer
- **The maintenance cost for this duplication is *far less* than the cost of maintaining the zero-duplication, complex, nested code.**
- **It is also faster to write it**
- **Simpler templates are more agile and can be adapted to changing needs more quickly… like improved SEO design**

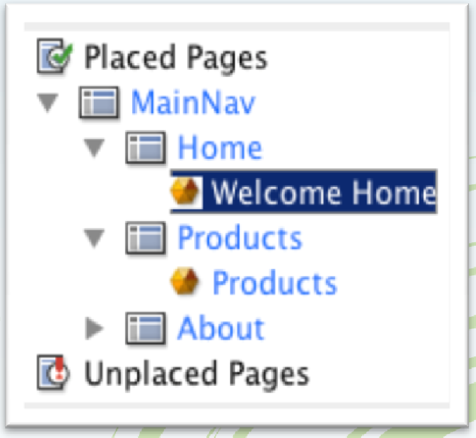
Designing Templates for SEO



```
<%@ taglib prefix="cs" uri="futuretense_cs/ftcs1_0.tld" %>
<%@ taglib prefix="ics" uri="futuretense_cs/ics.tld" %>
<%@ taglib prefix="render" uri="futuretense_cs/render.tld" %>
<%@ page import="com.fatwire.assetapi.data.AssetData,
                 com.fatwire.gst.foundation.facade.assetapi.AssetDataUtils,
                 com.fatwire.gst.foundation.facade.assetapi.AttributeDataUtils,
                 com.fatwire.gst.foundation.taglib.NavigationHelper" %>

<cs:ftcs>
    <ics:if condition='<%=ics.GetVar("tid")!=null%>'><ics:then><render:logdep cid='<%=ics.GetVar("tid")%>' c="Template"/></ics:then></ics:if>
    <%
        AssetData ad = AssetDataUtils.getAssetData(ics.GetVar("c"), ics.GetVar("cid"),
                "metatitle", "metadescription", "metakeyword", "h1title", "linktext", "body");
    %>
    <html>
    <head>
        <meta name="title" content='<%=AttributeDataUtils.asString(ad.getAttributeData("metatitle"))%>' />
        <meta name="description" content='<%=AttributeDataUtils.asString(ad.getAttributeData("metadescription"))%>' />
        <meta name="keyword" content='<%=AttributeDataUtils.asString(ad.getAttributeData("metakeyword"))%>' />
        <title><%=AttributeDataUtils.asString(ad.getAttributeData("metatitle"))%> | <ics:getvar name="site" /></title>
    </head>
    <body>
    <div id="nav">
        <render:satellitepage [my nav pagelet] ... />
    </div>

    <p>Current date is <strong><%= new java.util.Date()%></strong></p>

    <h1><%=AttributeDataUtils.asString(ad.getAttributeData("h1title"))%>
    </h1>

    <div class="articlebody">
        <%=AttributeDataUtils.asString(ad.getAttributeData("body"))%>
    </div>

    </body>
    </html>
</cs:ftcs>
```

Use Site Maps

- Site Maps help users to find pages
- Sitemap.xml files help search engines find pages
- Build your site map from your navigation structure
  - Then handle exceptions
- The Site Plan Tree is a great place to start
- Attach a vanity URL to the sitemap.xml to create a dynamic site map that does not require exporting


Placed Pages
▼ MainNav
▼ Home
Welcome Home
▼ Products
Products
▶ About
Unplaced Pages

Using Community Server to Improve Page Rank

- **If your site's content is not text-based, allow your users to write text for you**
  - UGC in the form of comments, if moderated, can be very effective
- **Community Server, added to pages, can add valuable keywords that help with page rank**
  - Example: YouTube

FATWIRE INTERACT 2010
GLOBAL USER CONFERENCE

Thank You

Tony Field
Global Solutions Architect
tony.field@fatwire.com