

WebGUI & Webware Users Community Workshop

Presenter: Corey White

Web Development & Marketing Consultant, donor.com

I. A Content Management Solution Mindset

- What is a CMS
- The benefits of CMS
- Why WebGUI?
- A Quick run-down of the UI

II. Building your WebGUI Site

- Setting up your site

- A. Assets Manager
- B. Admin Console (Settings, Users & Groups, Version Tags, Workflows, Spectre)
- HTML::Template and WG Macros**
- Building your site template hierarchy**
 - A. Page Style Templates
 - B. Page Layout Templates
 - C. Content Templates (articles, snippets, CS)

A Content Management Mindset

What is a Content Management System?

A web content management system is a software system used to manage and control a large, dynamic collection of web material (HTML documents and their associated images). A CMS facilitates document control, auditing, editing, and time line management. A Web CMS provides the following key features:

1. **multiple tiered management** with separation of design, structure & content
2. a **graphical user interface** that interacts with the user's permissions and abilities
3. **decentralized** maintenance
4. designed with **non-technical content authors** in mind
5. Configurable **access restrictions** & site **visitor segmentation**
6. Consistency of **design is preserved, templates provide global control**
7. Template driven for **ease-of-use**.
8. Content is stored in a **database** (can re-use and manipulate)
9. Integration of **web modules and plug-ins** (wikis, collaboration systems, blogs, photo galleries, calendar, etc.)
10. Compliance with **web standards** and regular updates that include new features sets and keep the system up to current web standards

Why Did Donorware/donor.com choose WebGUI as it's CMS platform?

Why did donor.com choose WebGUI?

1. **Open Source CMS** (software is free - no license fees – only expense is support)
2. Runs in **PERL and on LINUX and MySQL**
3. Browser based software that runs on **all OS platforms**
4. **Full featured** Content Management System and web framework
5. Plainblack (WebGUI's creator) **shares donor.com's philosophies** of open source
6. Very **tight collaboration** – donor.com wrote and contributed the WSCClient module and has contributed other changes that are now live
7. **Extensible** – can get under the hood and add new capabilities
8. EVERYTHING is template driven, giving lots of control over “look & feel” and function.



Admin Console

Asset Control Tabs

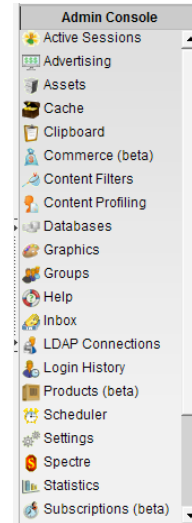


Assets Manager



Page Layout Edit

Admin Console – administrative functions over the site (users, display, cache, authentication)



New to WebGUI 7.4 (more on these later)

Spectre
Version Tags
Workflow

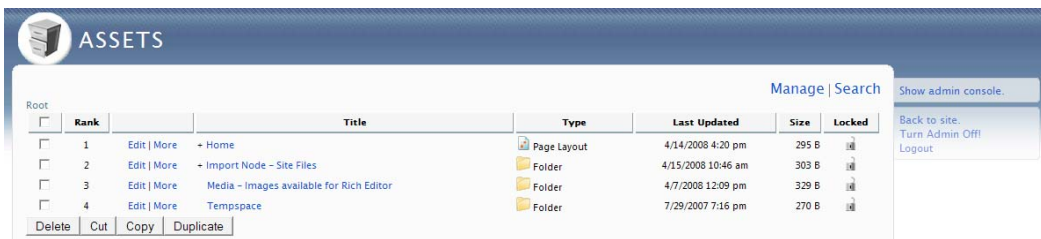
Assets Manager –

Import Node: storage of all site collateral to be used in generating the site (this can be templates, files, images, HTML/CSS/JS/XML docs and assets).

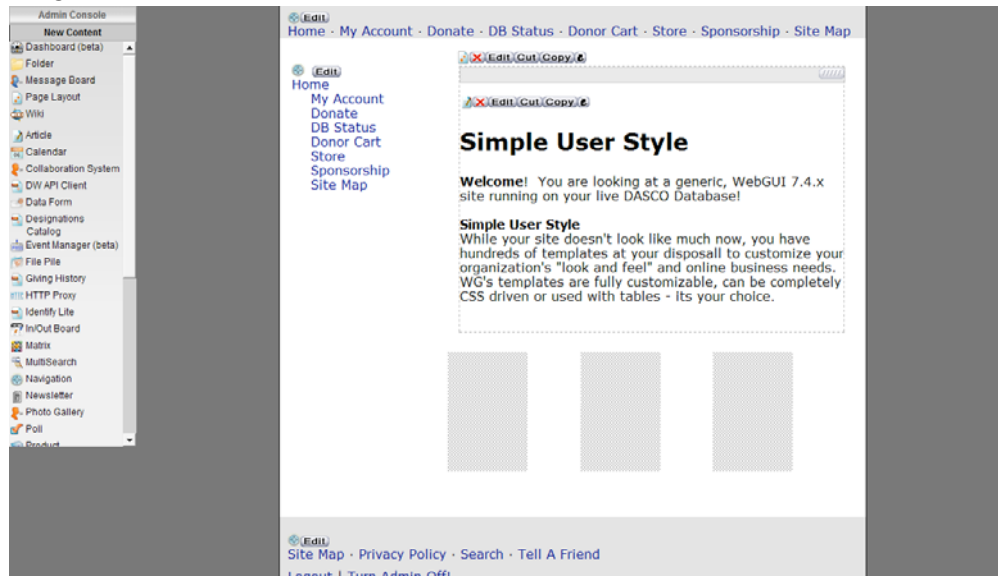
Page Tree: Asset used in the site. Displayed as a hierarchy

Media Folder: Storage of media uploaded via Rich Editor

Tempspace: Storage of media uploaded via the wiki



Page View Edit (Admin On)– manage web pages in a “live” view



4/15/2009

donor.com community conference

11

Asset Control Tabs

Properties Display Security Metadata save cancel

Properties

Title
Menu Title
URL
Description

Display

Display options
Navigation Settings
Contextual
Templates

Security

Owner
View Perms
Edit Perms

Meta Data

Synopsis (summary)
Head Elements
Package
Prototype

4/15/2009

donor.com community conference

12

Every Asset type in WebGUI has an asset control tab



New to WebGUI 7.4 (more on these later)

- Calendar Upgrade
- Data form upgrade
- Even Manager
- Multi-search and Search Upgrade
- Newsletter
- Project Manager
- SQL Form
- Wiki

New Content

- Dashboard (beta)
- Folder
- Page Layout

- Article
- Calendar
- Collaboration System
- Data Form
- Event Manager (beta)
- File Pile
- HTTP Proxy
- In/Out Board
- Matrix
- Message Board
- MultiSearch
- Navigation
- Newsletter
- Photo Gallery [Edit](#)
- Poll
- Product
- Project Manager (beta)
- Redirect
- Request Tracker [Edit](#)
- SQL Form (beta)
- SQL Report
- Search
- Snippet
- Stock Data
- Survey
- Syndicated Content
- Time Tracker (beta)
- WeatherData
- Web Services Client
- Wiki
- Zip Archive

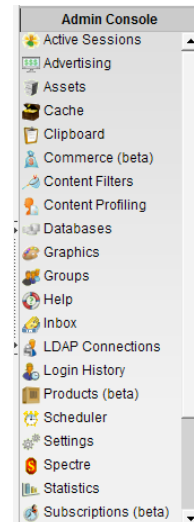
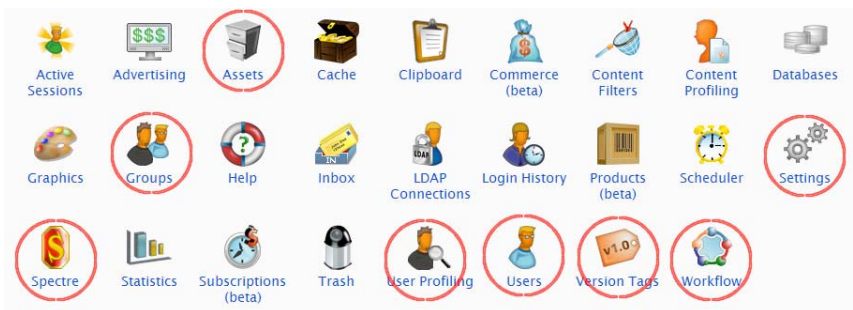
- File
- Image
- Rich Edit
- Template

Setting Up Your Site

Admin Console

Setup: Admin Console

Setting Up Your Admin Console





Configuring your Site Settings



Default view of settings

- company name: ^c(); used in content will display this name
- company email: ^e(); used in content will display email
- company URL: ^u(); used in content will display URL

*^ used in macros. More on macros later



Hovering your mouse over the field title will display a description of that setting.

Company	Content	UI	Messaging	Miscellaneous	User	Authentication	Permissions	save	cancel
Default Version Tag Workflow	Commit Without Approval							Manage	
Automatically request commit?	<input type="radio"/> Yes <input checked="" type="radio"/> No								
Skip commit comments?	<input type="radio"/> Yes <input checked="" type="radio"/> No								
Trash Workflow	Do Nothing on Deletion							Manage	
Purge Workflow	Do Nothing on Deletion							Manage	
Change URL Workflow	Do Nothing on Deletion							Manage	
Default Home Page	Home								
Not Found Page	Home								
URL Extension									
Maximum Attachment Size	10000								
Max Image Size	100000								
Thumbnail Size	50								
Enable Metadata?	<input type="radio"/> Yes <input checked="" type="radio"/> No								



You can by-pass the versioning system completely by selecting yes for "Automatically request..." and "Skip".

You can still create custom version in the future. Example – create a version for a new site area.

Versions explained later....

Company	Content	UI	Messaging	Miscellaneous	User	Authentication	Permissions	save	cancel
Default Rich Editor	Content Manager's Rich Edit								
Text Box Size	30								
User Function Style	Style 03							Edit/Manage	
Admin Console Template	Admin Console							Edit/Manage	

Show admin console.

Donorware / 4/9/2008 12:55 pm (Autotag)

Back to site.
Turn Admin Off!
Logout

Rich Edit – Default global "text" editor to use in content.

User Function Style – style to use for Login template

Admin Console Template – style used to display template



You can create multiple rich edit types and customize how they are displayed within content at ...root/import/richedit?func=manageAssets



DO NOT select a new Admin Console template unless you have created one that uses all the required variables. Doing so will disable the Admin console

Control email settings for internal user-to-user emails.

Enable debugging and set host properties.



Put in your own IP to see debugging for your IP only! WG debug allows you to see all of the WebGUI and Webware processes running on your site!

4/15/2009

donor.com community conference

23

4/15/2009

donor.com community conference

These settings come pre-configured on your donor.com WG site.

It is recommended you do not change these settings without consulting donor.com



WebGUI sites hosted with donor.com are uniquely configured and built to directly interface with DASCO via webware.

Setting up webware pages, configuring the APIs and customizing the webware templates are all done utilizing your webgui site and UI.

24

SETTINGS

Company Content UI Messaging Miscellaneous User Authentication **Permissions** save cancel Show admin console.

Setting	Permission	Action
Active Sessions	Admins	Manage
AdSpace	Admins	Manage
Cache	Admins	Manage
Commerce	Admins	Manage
Cron	Admins	Manage
Database Link	Admins	Manage
Graphics	Admins	Manage
Groups	Admins	Manage
Groups (limited)	Secondary Admins	Manage
Help	Everyone	Manage
LDAP	Admins	Manage
Login History	Admins	Manage
Products	Product Managers	Manage
User Profiling	Admins	Manage
Content Filters	Admins	Manage
Spectre	Admins	Manage
Statistics	Admins	Manage
Subscriptions	Admins	Manage
Transactions	Admins	Manage
Users	Admins	Manage
Users (add only)	Secondary Admins	Manage
Version Tags	Turn Admin On	Manage
Workflow	Workflow Managers	Manage
Workflow (run)	Admins	Manage

Donorware / 4/9/2008 12:55 pm (Autotag)
Back to site.
Turn Admin Off!
Logout

Setting Up Groups

Steps to successfully configuring your groups

1. Understand the “default” WebGUI groups and their implications

Group Name	Description	User Count
Ad Manager	These users will be able to manage advertisements.	0
Admins	Anyone who belongs to this group has privileges to do anything and everything.	1
Content Managers	Users that have privileges to edit content on this site. The user still needs to be added to a group that has editing privileges on specific pages.	0
Export Managers	Users in this group can export pages to disk.	0
Package Managers	Users that have privileges to add, edit, and delete packages of wobjects and pages to deploy.	0
Product Managers	The group that is allowed to edit, delete and create products.	0
Secondary Admins	Users that have limited administrative privileges.	0
Template Managers	Users that have privileges to edit templates for this site.	0
Turn Admin On	These users can enable admin mode.	0
Version Tag Managers	People who can create, edit, and delete special version tags.	0
Workflow Managers	People who can create, edit, and delete workflows.	0

A group not shown here is “registered users”. This group is automatically assigned to all new users who register via identify/multi-method.

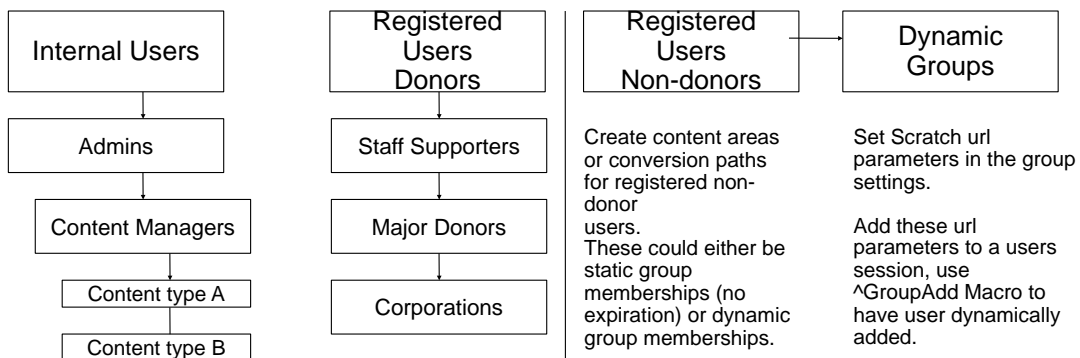


If creating custom content groups, precede your group name with a suffix like “CG” (for content group) or another name so that you can easily differentiate site administration groups from site user groups.

4/15/2009

donor.com community conference

2. Put together a group schema for your organization



DASCO Special Lists can drive WebGUI groups to control site access

4/15/2009

donor.com community conference

A Site Group Schema needs to....

1. Consider all constituent types (internal & external)
2. Identify which types will be DASCO driven and which will be WebGUI Driven.
3. Needs to be documented in your group description and preferably your group name as well.

3. Create your Groups

Group ID – WebGUI creates a unique ID for groups. These can be used in the Group Macros.

Group Name – name for Group

Description – what's the group for!

Expire Offset – time before the group expire

Expire Notification – how much advance notice a group member will receive before their group membership expires

Delete Offset – number of days from notification and actual expiration

IP Address – Any user IP address that manages these IP(s) will be added to this group. INTRANET!

Scratch Filter – Binds users dynamically to a group by a scratch variable in their session

Add/Remove – let users remove themselves from a group

Database Link – validate users against an external db

SQL Query – to validate against a user db, you may construct an SQL statement that will return a list of WebGUI user ids for users in this group **BE VERY CAREFUL BEFORE EXECUTING THIS**

LDAP – bind users in a group to an existing LDAP directory

4. Add users to your manual groups



If groups are more than for creating site edit and administrative permissions, what are other scenarios you could use for groups?

- Pass a scratch filter through an inbound campaign url. Segment your marketing visitors into different groups and then use the ^GroupText macro or security settings on articles to change the offers based upon the group

- Use IP for INTRANET, vendors, etc. HINT – if you view the login history, you can get the IP address of any user who came to your site.

5. Employ your Group schema while you build out the site.

Setting Up Users

1. **WebGUI Users** – create users using the standard WebGUI user creation interface
2. **WebGUI/DASCO Users** – syncprofile handles this

DO NOT CONFIGURE DASCO USER SETTINGS IN WEBGUI



User Profiling

Allows you to store custom user fields in WebGUI to “re-use” in content.... dynamic personalization!

- ✓ You need to create a **custom output spec** to determines what fields are passed to webgui

Custom Output Spec - standard dasco feature that allows you to customize your data output into any format.



WebGUI Version Tags

1. Version tags allow you to save all your site work under a session

- A. WebGUI begins a new version session with each login – version begins once you edit an asset in webgui
- B. You can create your own version within the Version Tag management screen

[stop – let's look at this together]

2. When you are ready to commit your work, commit your version tag.

3. If you log out before you commit the version, WG will save it for you.



WebGUI Version Tags

4. If a version remains uncommitted, the assets worked on will be lock until you commit it.

[stop – let's commit a version]

5. Once a version it committed, it can be reviewed, rolled back or deleted.

[stop – let's review, roll-back and delete]

6. You can bypass versioning all together at the settings:content screen
7. You can create a custom work-flow for how versions are committed.



WebGUI Workflows

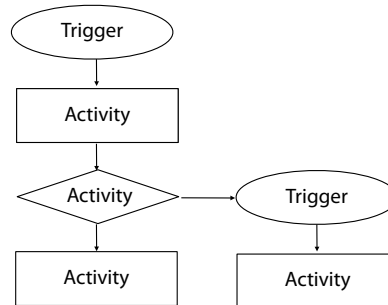
We will only review these briefly so that they are understood conceptually.

What is a workflow?

An event triggered state machine and execution system.

Parts of a workflow engine

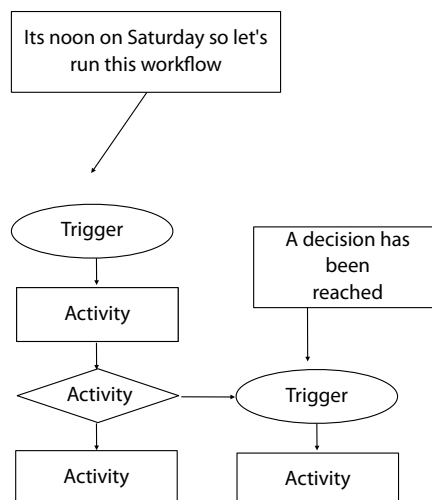
- ✓ **Triggers**
- ✓ **Workflows**
- ✓ **Activities**



Triggers

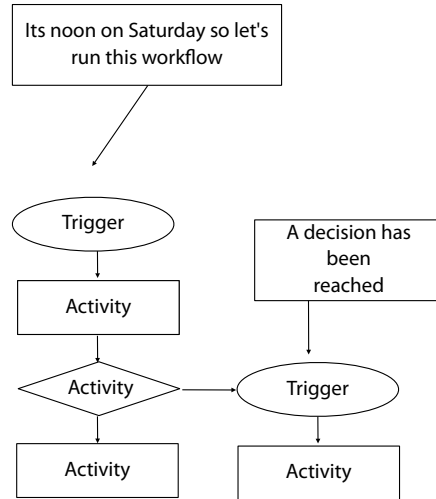
Events that cause a workflow to execute

- ✓ It's the right time in the right place
- ✓ Somebody did something



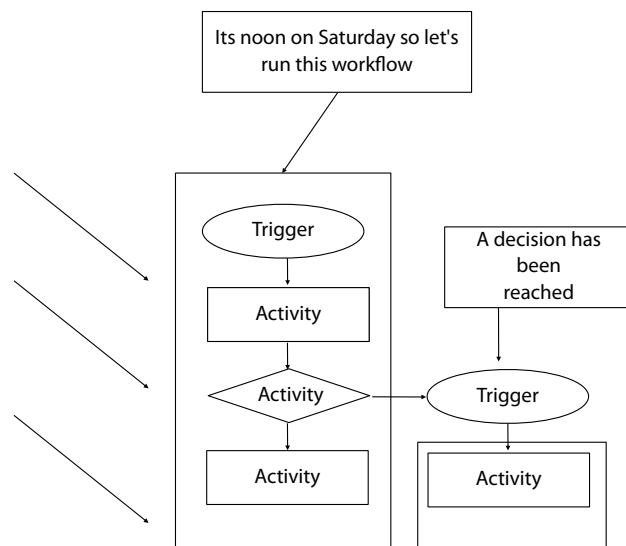
WorkFlows

- ✓ Chains of tasks to be executed
- ✓ Sometimes called an execution chain



Activities – Do Something

- ✓ Run an external program
- ✓ Ask a user for input
- ✓ Publish some content



Synchronous

- ✓ Next activity executes directly after the previous
- ✓ Blocks while waiting on external input

✓ **Example:**

Apple Automater

WebGUI 7.4+

Asynchronous

- ✓ Next activity executes when controller says so.
- ✓ Does not block while waiting on external input

✓ **Example**

WebGUI Spectre

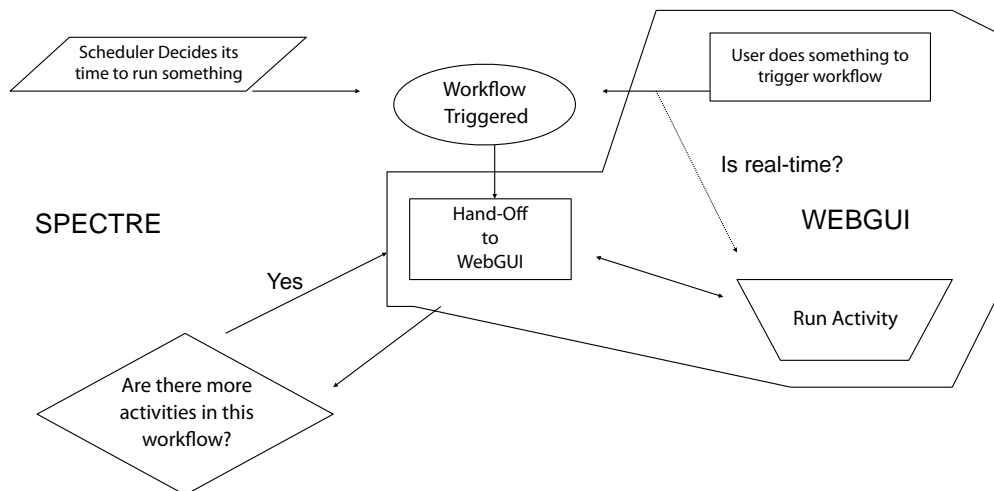


WebGUI Spectre

WebGUI Spectre

Supervisor of perplexing event-handling contraptions for triggering relentless executions

Workflow Engine Overview



What you need to know?

- ✓ Is spectre running?
- ✓ If so, what workflows?
- ✓ What tasks are scheduled.



If you commit a version, but don't see it pushed live to the site, this could be that spectre is either not running... or your workflow is in line behind other workflows that need to run.



Help: Advice on how to use it?

JUST USE IT!!!

- ✓ **Macros – list of available and how to use them**
- ✓ **Templates – list of variables available in templates**

Macros and Templates are the 2 great strengths of building your sit

A successful CMS administrator must know these!



The Assets Manager

3 Main branches – Import Node, Page Tree, Media

								Manage Search
<input type="checkbox"/>	Rank		Title	Type	Last Updated	Size	Locked	
<input type="checkbox"/>	1	Edit More	home	Page Layout	2/19/2008 11:20 am	451 B	<input type="checkbox"/>	
<input type="checkbox"/>	6	Edit More	+ Import Node	Folder	12/19/2007 3:09 pm	272 B	<input type="checkbox"/>	
<input type="checkbox"/>	7	Edit More	Media	Folder	5/14/2006 4:33 pm	258 B	<input type="checkbox"/>	
<input type="checkbox"/>	9	Edit More	Tempspace	Folder	11/15/2007 3:49 pm	270 B	<input type="checkbox"/>	

Page Tree
 - Organization of live site assets
 - Assets are placed by hierarchy
 - Assets placed on a live, non-hidden page will be displayed

Media
 - "storage" for assets (mostly images) available to site via content managers rich edit

Import Node
 - "storage" for assets "to-be-used"
 - assets can be sorted via WG macros (^AssetProxy, ^FileUrl, ^RandomAssetProxy)
 - assets can be sorted with standard html types

***Tempspace** - It is used for uploads when editing wiki articles. The uploads are moved to a more permanent location when saving the edits. It automatically gets emptied once per day, so you wouldn't want to leave anything in it that you needed to keep.

99	Edit More	+ Images	Folder	2/28/2008 1:12 pm	310 B	
100	Edit More	+ Files	Folder	9/28/2007 4:46 pm	277 B	
101	Edit More	+ Snippets	Folder	9/28/2007 4:46 pm	286 B	
102	Edit More	Prototypes	Folder	4/1/2008 4:39 pm	355 B	
104	Edit More	Assets for Proxy	Folder	4/1/2008 4:39 pm	373 B	
105	Edit More	+ Styles, Layouts & Article Templates	Folder	4/1/2008 3:21 pm	389 B	
106	Edit More	+ Custom Asset Templates	Folder	4/1/2008 3:25 pm	391 B	
107	Edit More	+ 3rd Party Add-ons	Folder	4/1/2008 3:19 pm	353 B	
108	Edit More	+ Navigation Configurations & Templates	Folder	4/1/2008 4:38 pm	435 B	
109	Edit More	+ WebGUI Content Engine Files (DO NOT EDIT)	Folder	4/1/2008 3:22 pm	384 B	



Create a folder for nav, style/layout/article templates, snippets, images, prototypes, 3rd party packages and add-ons. Store ALL custom templates in here.

Create a WebGUI folder and place all defau

l
t templates in this folder (NEVER EDIT THESE FILES)

Some tips within the Assets Manager

Search (new) – now search for assets within the assets manager

“**more**” - change url, edit branch, short-cut, revisions, lock



Change URL – always use WG's change url feature when renaming a url

Edit Branch – apply master page configurations to page and all it's descendants with the edit branch function.

BREAK

WG BASICS

- I. HTML::Template Basics
- II. WebGUI Macros
- II. CMS Hierarchy
 - A. Page Style Template
 - B. Page Layout Template
 - C. Asset Templates

What in WebGUI is driven by templates?

What in WebGUI is NOT driven by templates?

EVERYTHING IN WEBGUI IS DRIVEN BY TEMPLATES!

- CMS function templates
- UI templates
- Design Templates
- Module Templates
- If it's in WebGUI it has a template!

Templates are your best friend!

Templates allow

1. Mass production of content
2. Style standardization
3. Separation of concerns
4. Ease of design change, ease of interface localization, multiple skill levels working together

What you need to know to use WebGUI's template system

1. WebGUI has a working prototype of every template that you will need in webbing. You can use WebGUI's template "as is" or make a copy of WebGUI's template and edit it.
2. WebGUI templates use standard **HTML or XHTML** (your choice), **CSS** and **HTML::Template Language**, and **WebGUI Macros**.
3. WebGUI templates are completely separated from the design and content. These templates "power" the every day functions a content manager does... yet the content manager will never have to interact with the template.

WG Basics: HTML::Template

The Basic tags used in an HTML::Template

TMPL_VAR

TMPL_LOOP

TMPL_IF / TMPL_ELSE/ TMPL_UNLESS

WEBSITE HELPS

sourceforge.net - http://html-template.sourceforge.net/html_template.html#the%20tags

perlmonks.org - http://www.perlmonks.org/?node_id=65642

SOME FACTS:

Utilizes **standard html**

You do not need to know how to write PERL.

HTML::TEMPLATE allows you to "manipulate" what and how data is displayed in the browser.

HTML::Template - TMPL_VAR

TMPL_VAR <TMPL_VAR NAME="PARAMETER_NAME">

example : <input name=param type=text value="<TMPL_VAR NAME="PARAM">">

Explanation:

The <TMPL_VAR> tag is very simple. For each <TMPL_VAR> tag in the template you call (PERL Script) \$template->param(PARAMETER_NAME => "VALUE").

When the template is output the <TMPL_VAR> is replaced with the VALUE text you specified. If you don't set a parameter it just gets skipped in the output.

WebGUI Syntax

WebGUI templates use a short-hand, <tmpl_var varName> (where varName is the variable name)

A list of ALL the variables that drive each template are available in WebGUI help!!!

HTML::Template - TMPL_LOOP

TMPL_LOOP <TMPL_LOOP NAME="LOOP_NAME"> ... </TMPL_LOOP>

In the temp

```
<TMPL_LOOP NAME=EMPLOYEE_INFO>
Name: <TMPL_VAR NAME=NAME> <br>
Job: <TMPL_VAR NAME=JOB> <p>
</TMPL_LOOP>
```

In the script

```
$template->param(EMPLOYEE_INFO => [
    { name => 'Sam', job => 'programmer' },
    { name => 'Steve', job => 'soda jerk' },
    ]
);

print $template->output();
```

HTML::Template - TMPL_LOOP

TMPL_LOOP <TMPL_LOOP NAME="LOOP_NAME"> ... </TMPL_LOOP>

The output in a browser:

Name: Sam

Job: programmer

Name: Steve

Job: soda jerk

HTML::Template - TMPL_IF

TMPL_IF <TMPL_IF NAME="PARAMETER_NAME"> ... </TMPL_IF>

Example:

```
<TMPL_IF NAME="BOOL">  
  Some text that only gets displayed if BOOL is true!  
</TMPL_IF>
```

HTML::Template - TMPL_ELSE

TMPL_ELSE <TMPL_IF NAME="PARAMETER_NAME"> ... <TMPL_ELSE> ... </
TMPL_IF>

Example:

```
<TMPL_IF BOOL>  
  Some text that is included only if BOOL is true  
<TMPL_ELSE>  
  Some text that is included only if BOOL is false  
</TMPL_IF>
```

HTML::Template - TMPL_UNLESS

TMPL_UNLESS <TMPL_UNLESS NAME="PARAMETER_NAME"> ... </TMPL_UNLESS>

Example:

```
<TMPL_UNLESS BOOL>
```

Some text that is output only if BOOL is FALSE.

```
<TMPL_ELSE>
```

Some text that is output only if BOOL is TRUE.

```
</TMPL_UNLESS>
```

WG Basics: Macros

WebGUI macros? ^X());

- ✓ WebGUI macros allow you to create dynamic content within otherwise static content.

For instance, you may wish to show which user is logged in on every page, or you may wish to have a dynamically built menu or crumb trail.

- ✓ Macros always begin with a caret (^) and follow with at least one other character and end with a semicolon (;). Some macros can be extended/configured by taking the format of ^X("config text");. When providing multiple arguments to a macro, they should be separated by only commas:
- ✓ You may nest Macros, but only if they do not contains commas or quotes. You should follow the following guidelines:
 - ✓ Macros which contain HTML or javascript cannot be nested.
 - ✓ Macros which return user input that may contain quotes or commas may be nested, but it's dangerous.
 - ✓ If your user input must contain quotes and/or commas, it is recommended that the HTML encodings for comma and quote be used.



http://web101.dwvvpn.net/db_status Shows a status test on some Macros
<http://web101.dwvvpn.net/?op=viewHelpIndex> Provides Macro help

AdminBar	<code>^AdminBar();</code>	RecentDiscussion	<code>^RecentDiscussion();</code>
AdminText	<code>^AdminText();</code>	International	<code>^International();</code>
AdminToggle	<code>^AdminToggle();</code>	L_loginBox	<code>^L();</code>
At_username	<code>^@();</code>	LoginToggle	<code>^LoginToggle();</code>
CanEditText	<code>^CanEditText();</code>	EditableToggle	<code>^EditableToggle();</code>
D_date	<code>^D();</code>	GroupDelete	<code>^GroupDelete();</code>
DisplayParam	<code>^DisplayParam();</code>	Page	<code>^Page();</code>
Env	<code>^Env();</code>	PageTitle	<code>^PageTitle();</code>
Extras	<code>^Extras();</code>	LastModified	<code>^LastModified();</code>
FileUrl	<code>^FileUrl();</code>	RootTitle	<code>^RootTitle();</code>
FormParam	<code>^FormParam();</code>	W3	<code>^W3();</code>
GroupAd	<code>^GroupAdd();</code>	Slash_gatewayUrl	<code>^/();</code>
GroupTxt	<code>^GroupText();</code>	AssetProxy	<code>^AssetProxy();</code>
H_homeLink	<code>^H();</code>	Spacer	<code>^Spacer();</code>
Hash_userid	<code>^#();</code>	Thumbnail	<code>^Thumbnail();</code>

PageUrl	^PageUrl();	WebGUIVersion	^WebGUIVersion();
User	^User();	WebGUIStatus	^WebGUIStatus();
a_account	^a();	SubscribeNow	^SubscribeNow();
c_companyName	^c();	BuyNow	^BuyNow();
e_companyEmail	^e();	Buy	^Buy();
GoodyDate	^GoodyDate();	Checkout	^Checkout();
AdSpace	^AdSpace();	Featured	^Featured();
r_printable	^r();	RegisteredUsers	^RegisteredUsers();
u_companyUrl	^u();	RandomThread	^RandomThread();
RandomAssetProxy	^RandomAssetProxy();	Karma	^Karma();
Product	^Product();	DBSchema	^DBSchema();
ImageOr	^ImageOr();		
SubscriptionItem	^SubscriptionItem();		
SubscriptionItemPurchaseUrl	^SubscriptionItemPurchaseUrl();		

The CMS Hierarchy

Page Style Templates

Page Layout Templates

Assets (content) Templates

Page Style Templates

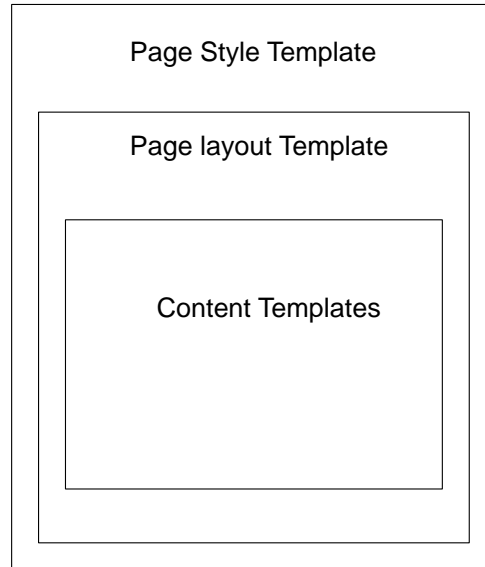
- Doc Header (CSS, JS, Snippets)
- Body Header (Main Navigation)
 - Footer

Page Layout Templates

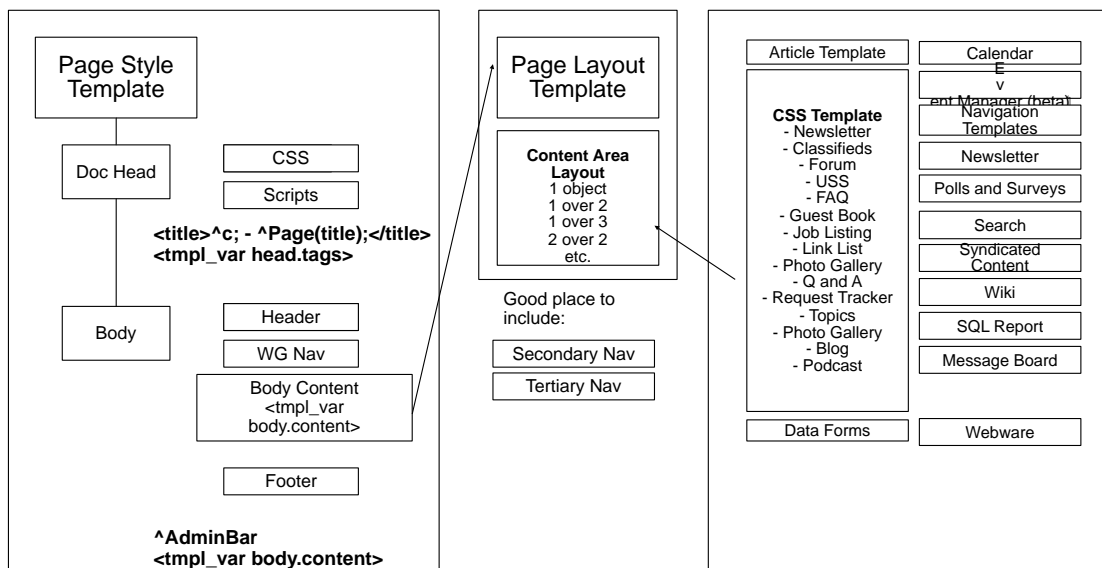
- Secondary & Tertiary Navigation

Content Templates

- Simple Content (article)
- Dynamic Content (CS)
- Forms (Data Forms)
 - Folder Gallery
 - Photo Gallery



The CMS Template Hierarchy



Page Style Template

- ✓ **Highest Level (design) template**
- ✓ **The main “look and feel” for the site**
 - ✓ **Holds all HTML Doc Header element**
 - ✓ **Usually holds the design header and main navigation**
 - ✓ **Incorporates WebGUI <tmpl_vars...> and macros**

```
<!--##### Declare any doc type HTML 4.0/strict/transitional, XHTML strict/transitional -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<!--##### Use WG Variables for Dynamic title-->
    <title>^Page(title); - <tmpl_var session.setting.companyName></title>
<!--##### Create a snippet for universal head elements (favicon, tracking code, etc)-->
    ^AssetProxy(universal-header-include);
<!--##### IMPORTANT variable used to pull required WG tags in header -->
    <tmpl_var head.tags>
<!--### CSS Reference - build CSS as Snippet and reference as normal-->
<link href="/simpleUser.css" rel="stylesheet" type="text/css" />
</head>
```

```
<body>^AdminBar;
<div id="container">
  <div id="header">
    <h1>^Page(title);</h1> ^AssetProxy(currentmenuhorizontal);
  </div>
  <div id="sidebar1">
    <p>^AssetProxy(currentmenuvertical);</p>
  </div>
  <div id="mainContent">
    <p><tmpl_var body.content></p>
  </div> <br class="clearfloat" />
  <div id="footer"> <p>^AssetProxy(footermenuhorizontal);</p> <p>^LoginToggle(); | ^AdminToggle(); </div></div>
</body>
</html>
```

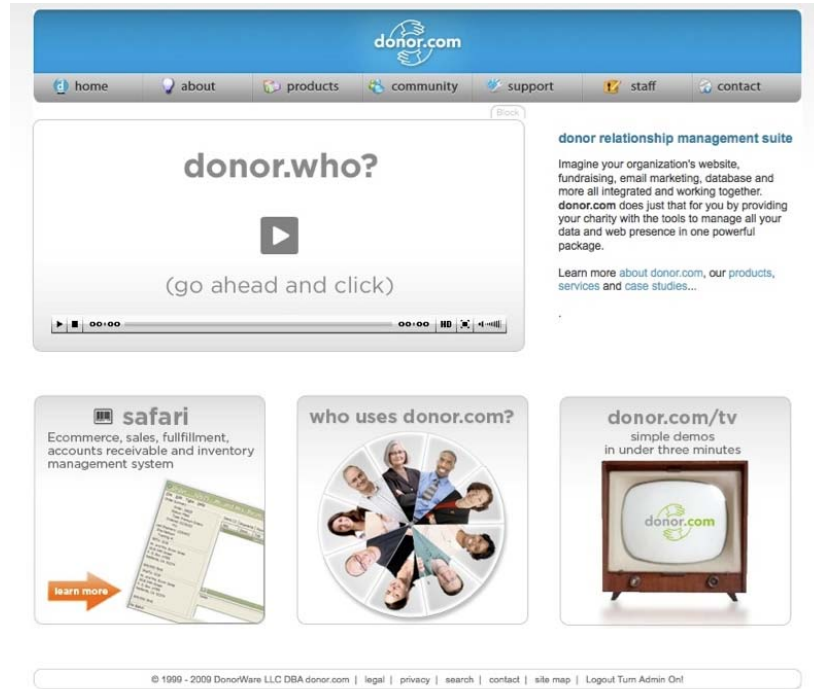
Macro that makes the Admin Bar appear in Admin Mode

Proxies in a Nav. Config Asset

This is the variable that references your page layout template

Macros to display the login link and the "Admin on/off"

Page Layout Templates



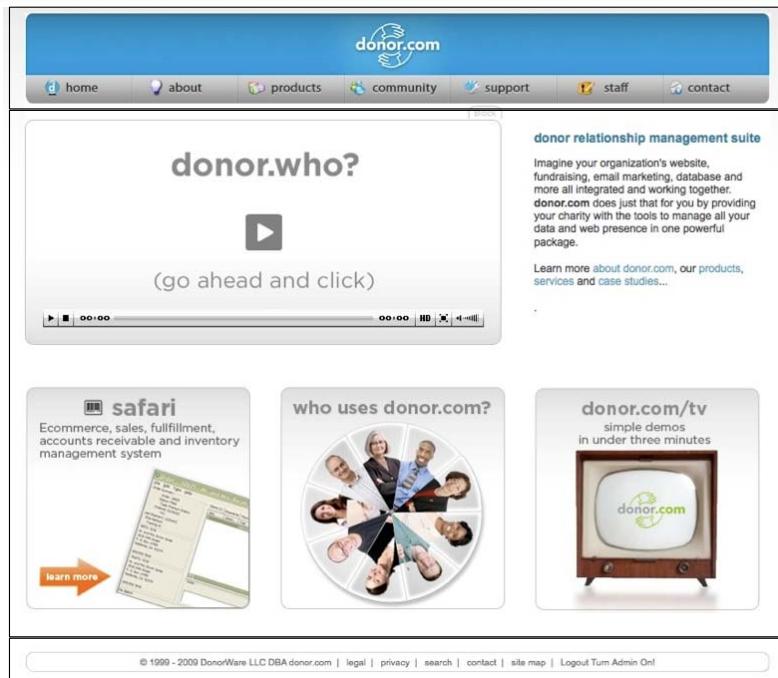
4/15/2009

donor.com community conference

75

Page Style Template

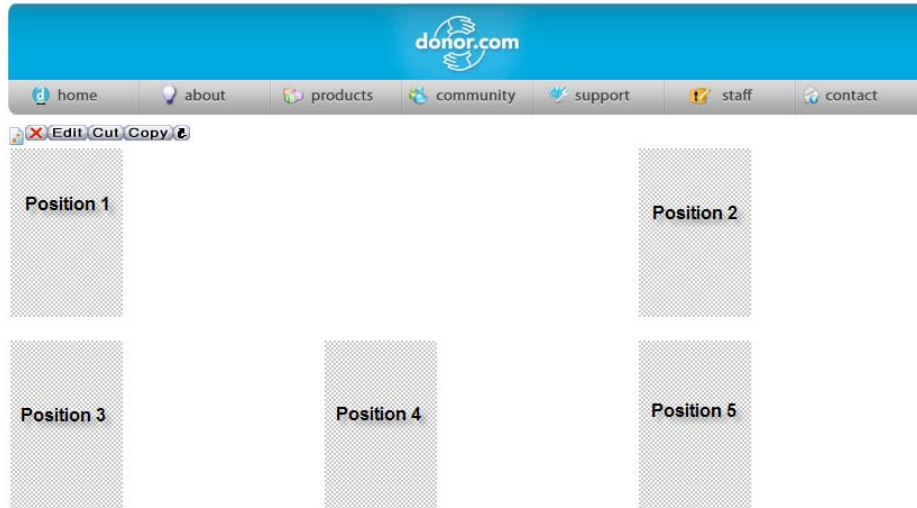
Page Layout Template



4/15/2009

donor.com community conference

76



This is a "raw" page layout in Admin View
Each position is a "content position place holder"

How do we get this?

Take your standard HTML

```
<div id="bodyContent">
<div id="bodyTopLeft">Position 1</div>
<div id="bodyTopRight"> Position 2</div>
<br class="clearAll" />
<div id="bodyBottomLeft">Position3</div>
<div id="bodyBottomCenter">Position4</div>
<div id="bodyBottomRight">Position5</div>
<br class="clearAll" />
</div>
```

And add.. this to each position

```
<div class="layoutColumnPadding">
<tpl_if showAdmin> <table border="0"
  id="position1" class="content"><tbody></tbody></tpl_if>
<tpl_loop position1_loop><tpl_if showAdmin>
  <tr id="td<tpl_var id">
    <td><div id="td<tpl_var id>_div" class="draggable">
      </tpl_if>
    <div class="content"><tpl_var dragger.icon>
      <tpl_var content></div>
    </tpl_if showAdmin>
  </div></td></tr>
</tpl_if></tpl_loop>
<tpl_if showAdmin></tbody></table></tpl_if>
</div>
```

Just make sure you change the 'id="position"' lable to correspond with the position you would like that area to be.

NOTICE THE HTML::TEMLAGE TAGS!!

CMS Hierarchy: Page Layout Template

```
<div class="content"><tmpl_var dragger.icon>
</div> <tmpl_var content>

<tmpl_if showAdmin>
</div></td></tr>
</tmpl_if>

<tmpl_loop>
<tmpl_if showAdmin></tbody></table>
</tmpl_if>

</div> </div>

<br class="clearAll" />
<div id="bodyBottomLeft">

  <div class="layoutColumnPadding">
    <tmpl_if showAdmin>
      <table border="0" id="position3" class="content"><tbody>
        <tmpl_loop position3_loop><tmpl_if showAdmin>
          <tr id="td<tmpl_var id>">
            <td><div id="td<tmpl_var id>_div" class="draggable">
              </tmpl_if>
            <div class="content"><tmpl_var dragger.icon>
              <tmpl_var content>
            </div>
          </tr></tbody></table>
        </tmpl_loop>
      </tbody></table>
    </tmpl_if>

  </div>

<div id="bodyBottomCenter">
  <div class="layoutColumnPadding">
    <tmpl_if showAdmin>
      <table border="0" id="position4" class="content"><tbody>
        <tmpl_loop position4_loop><tmpl_if showAdmin>
          <tr id="td<tmpl_var id>">
            <td><div id="td<tmpl_var id>_div" class="draggable">
              </tmpl_if>
            <div class="content"><tmpl_var dragger.icon>
              <tmpl_var content>
            </div>
          </tr></tbody></table>
        </tmpl_loop>
      </tbody></table>
    </tmpl_if>

  </div>

</div>

<div id="bodyBottomRight">

  <div class="layoutColumnPadding">
    <tmpl_if showAdmin>
      <table border="0" id="position5" class="content"><tbody>
```

When you are done, you should have something that looks like this...

4/

79



CMS Hierarchy: Page Layout Templates

Review – setting up your page layout template

1. Take the html that will make up the page layout
2. Insert “the content position place-holder” code for each position
3. Last, add the remaining WG tags to your template

3. Last, add the remaining WG tags to your template

ASSET ID (pulls in the unique asset id for the template)

```
<div><a name="id<tmpl_var assetId>" id="id<tmpl_var assetId>"></a></div>
```

showAdmin (if admin on, show the “edit controls”)

```
<tmpl_if showAdmin>
<p><tmpl_var controls></p>
</tmpl_if>
```

displayTitle (if page is configured to display title)

```
<tmpl_if displayTitle>
  <h2><tmpl_var title></h2>
</tmpl_if>
```

description (if page has a description)

```
<tmpl_if description>
  <p><tmpl_var description></p>
</tmpl_if>
```

3. Last, add the remaining WG tags to your template

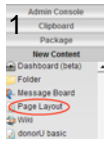
dragger.init (initiate the WG asset “drag-and-drop” feature)

```
<tmpl_if showAdmin>
  <table><tr id="blank" class="hidden"><td><div><div class="empty">&nbsp;</div></div></td></tr></table>
  <tmpl_var dragger.init>
</tmpl_if>
```



If you study the code for the “content position place-holders”, you'll see that that <tmpl_if showAdmin> is interdispersed.

This is how WG builds the graphical place-holder that's displayed when admin is on and uses the “lock” and “drag-drop” properties.



1. Select page layout to add a new page
2. Fill out the "properties" tab



- **Title** <tmpl_var pageTitle> the title of the page
- **Menu Title** <tmpl_var menuTitle> if shown in navigation, the title used
- **URL** <tmpl_var pageURL> the url for that page
- **Description** <tmpl_var description> - can either enter html content here or leave blank and insert an article to the page for content.



3. Complete the "display" tab
 - Decide if page will display in navigation
 - When page link clicked on, will open in new window?
 - Decide if title (<tmpl_var pageTitle> will be displayed

Style Template – Choose the Style Template the page uses

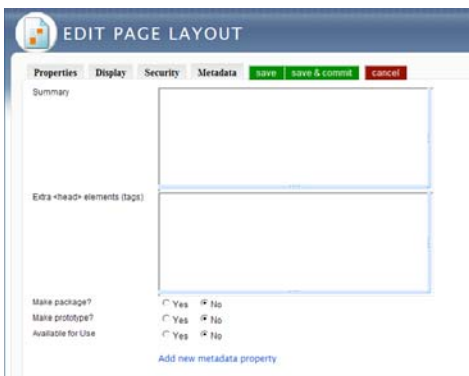
Page Layout Template – Choose the Page Layout template for this page

Printable – (if used) what is the style for "print page"



4. Fill out the security tab

- **Encrypt Content** – will enable SSL for page
- **Owner** – who is the page owner?
- **View** – set group viewing permissions
- **Edit** – set edit permissions



5. Meta Data Tag

- **Summary** – <tmpl_var synopsis> can pull into navigation templates
- **Extra Head Elements** – add extras to head
- **Package** – allows this asset and all its simblings to be exported.
- **Prototype** – makes a "prototype/image" of this asset so that it can be re-used again and again without having to configure all the tabs.
- **Available for use** – turn this page on or off

Asset Templates are the “meat” of WG templates. While page style and page layout templates control structure & design, asset templates give the website it's meaning, it's content.

Types of Asset/Content Templates

- 1. Simple (single function):** Asset powered by one single template (article template)
- 2. Multi/Dynamic Templates:** These assets are powered by multiple templates that server different functions in the asset's life-cycle.

Example: collaboration system template

- post template, thread template, module display template, search template and RSS template.

- 3. DB/API driven templates (webware):** These templates are the most complex (not necessarily in their difficulty, but in their function).

API driven templates are displayed to the end user by 2 sets of criteria

- A. The returns being passed through the API from DASCO, configured by the config DB.
- B. The end-user interaction
 - ✓ SOAP Transport Errors
 - ✓ Pre-population Mode
 - ✓ Method Errors
 - ✓ Successful Results

we'll cover DB/API webware templates in our afternoon session.

Pagination first page and image

```
<tmpl_if pagination.isFirstPage>
<tmpl_if image.url>
" style="padding:5px;" />
</tmpl_if>
</tmpl_if>
```

Description

```
<tmpl_if description>
<p><tmpl_var description></p>
</tmpl_if>
```

Pagination (last), Link, Attachment

```
<tmpl_if pagination.isLastPage>
<tmpl_if linkUrl>
<tmpl_if linkTitle>
<a href="<tmpl_var linkUrl>"><tmpl_var linkTitle></a>
</tmpl_if>
</tmpl_if>
<tmpl_if attachment.name>
<p style="display:inline;vertical-align:middle;">
<a href="<tmpl_var attachment.url>">" /
  >
<tmpl_var attachment.name></a></p>
</tmpl_if>
```

Pagination Variables

```
<tmpl_if pagination.pageCount.isMultiple>
<tmpl_var pagination.previousPage>&#183;
<tmpl_var pagination.pageList.upTo20>&#183;
<tmpl_var pagination.nextPage>
</tmpl_if>

<!--[end/]-->
```

Is this all that is available in an article? Let's check out the help docs 

http://web101.dwvnpn.net/?op=viewHelp:hid=article%20template:namespace=Asset_Article

new.template

Articles have the special ability to change their template so that you can allow users to see different views of the article. You do this by creating a link with a URL like this (replace 999 with the template Id you wish to use):

```
<a href="<tmpl_var new.template>999">Read more...</a>
```

description.first.100words/75words/50words/25words/10words

The first N words in the description. Words are defined as characters separated by whitespace, so HTML entities and tags count as words.

description.first.paragraph/second/third/fourth

description.first.sentence/first/second/third/fourth

session scratch variables

Snippets

Snippets are probably one of the most useful assets in WebGUI, but you need to know how and where to use them.

Whereas most assets have templates - snippets don't - they are what they are. Snippets don't have to be large either - they can be as simple as one character. And they're cached in WebGUI so they're pretty fast to work with.

Snippets are also versioned - so any changes you make to them won't be made until it has been committed.

Because of this simplicity, it's easy to overlook how useful they really are.

Snippets in Style Templates

```
^AssetProxy(header);  
^AssetProxy(css);  
^AssetProxy(javascript);  
^AssetProxy(body);  
^AssetProxy(navigation);  
^AssetProxy(body2);  
^AssetProxy(footer);
```

Snippets as variables

Snippets can be a variable in a WebGUI template or a Webware API template (more later).

Collaboration System Templates

A collaboration system is a versatile asset

- ✓ for sharing thoughts and discussion (forum), pictures (photo gallery), ideas (weblog) with other people.
- ✓ It creates a platform for people to share and exchange.
- ✓ for Content Managers to update their website in an easy way.

There are many default CS applications in Webgui

- ✓ But a CS can also be made into something different. With different settings, combination of variables and your own html and css, you can make from a default CS your own.
- ✓ Later examples, but let's first look at the different Collaboration Systems as default available in Webgui.

- forum
- weblog
- request tracker
- Q and A
- photo gallery
- classified
- FAQ
- topics
- (un)ordered list
- jobs
- guest book
- link list
- traditional with thumbnail

They all look different, but there are many similarities.

- ✓ All have a Collaboration system view (= overview), a thread view and a post view
- ✓ A post is added, there is no versioning
- ✓ As we will see later: variables and settings

CSS Asset Basic End-User Views

Overview

Thread

Post/Edit

Thread

RSS

Overview

Here you see an overview of all the posts



4/15/2009

donor.com community conference

101

Thread

Here you see an individual post with replies: the thread



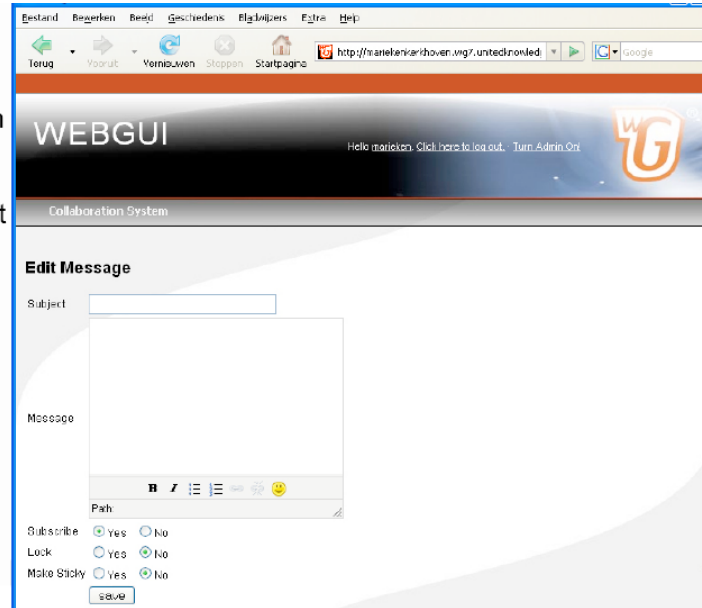
4/15/2009

donor.com community conference

102

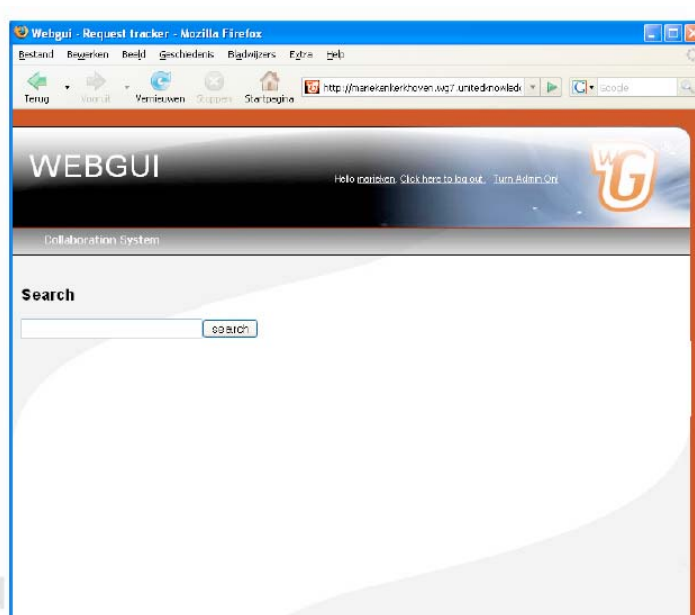
Post & edit

With the post form a new post is added or a post is edited.



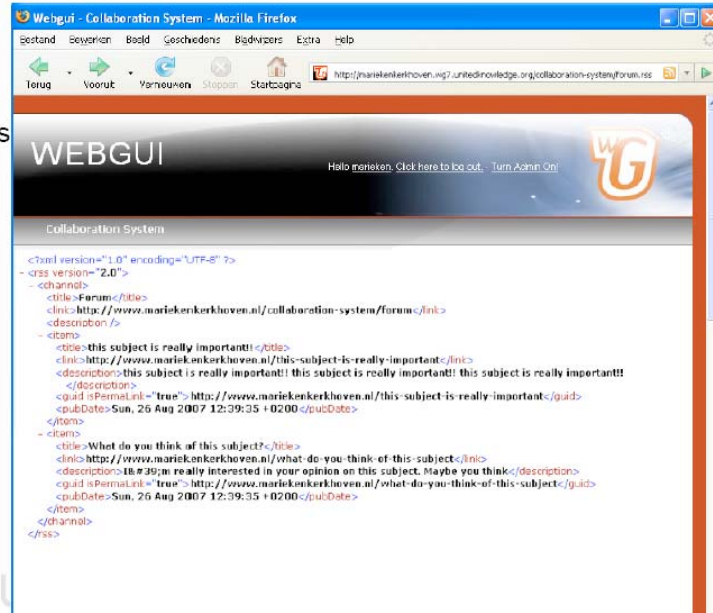
Search

You can search through a collaboration system for posts.



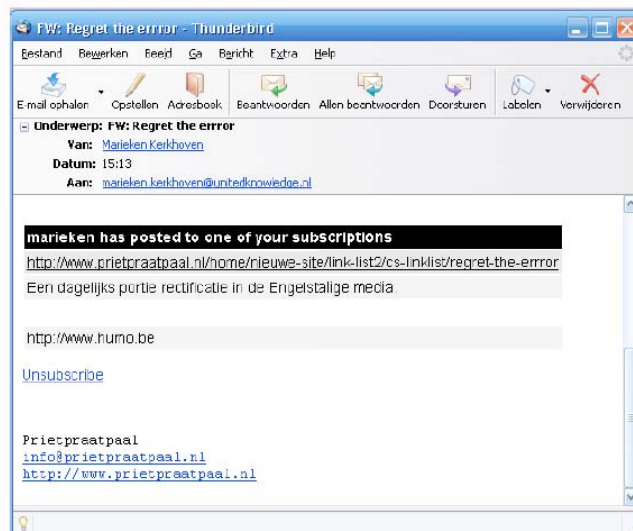
RSS

A RSS feed page is generated if in the settings (under display) 'enable RSS' is selected.



Mail

You receive an email in the mailbox you have registered as a user.



CMS Settings

- ✓ What do they do?
- ✓ And which variables do they generate? The variables will later also be seen in the templates.

Content/Assets Interface – PROPERTIES, DISPLAY, SECURITY, META DATA

As in other assets you have:

Asset ID: `<tmpl_var assetId>`

Title:

`<tmpl_if displayTitle>`
`<h2> <tmpl_var title> </h2>`
`</tmpl_if>`

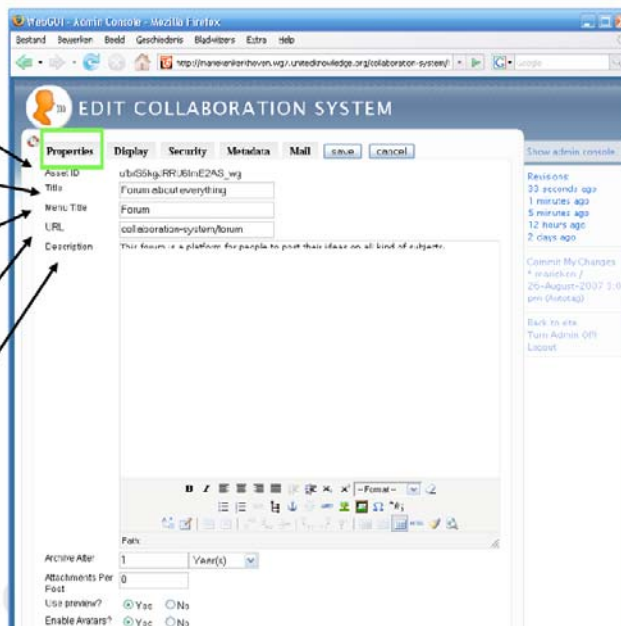
Menu title:

`<tmpl_var menuTitle>`
 as it will appear in your navigation

URL: `<tmpl_var url>`

Description:

`<tmpl_if description>`
`<tmpl_var description>`
`</tmpl_if>`



Next in the properties settings:

Archive after:

<tmpl_if archived>
 <tmpl_var archive>
 <tmpl_var unarchive>

Attachment per post:

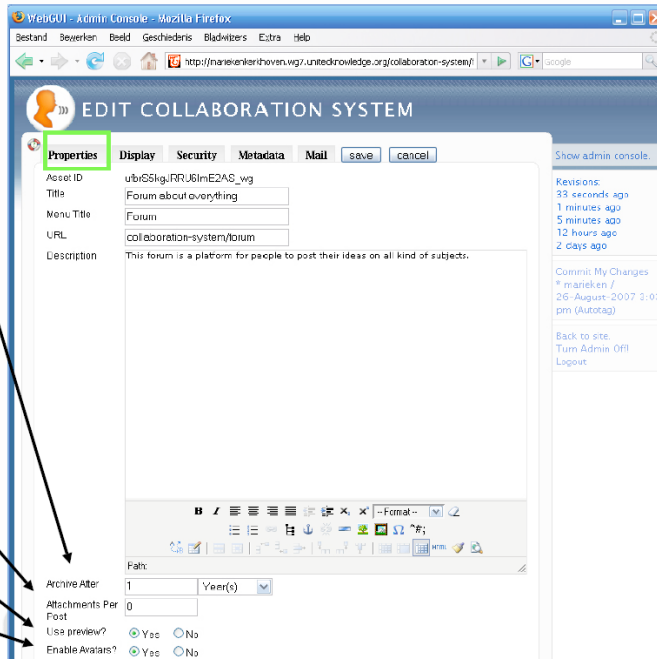
<attachment_loop> an attachment field is added in the post form.

Use preview:

in post form this will show up as:

<tmpl_if usePreview>

Enable avatar: <tmpl_var avatar.url>



4/15/2009

donor.com community conference

109

First some familiar settings:

Hide from navigation:

whether assets shows up in your navigation

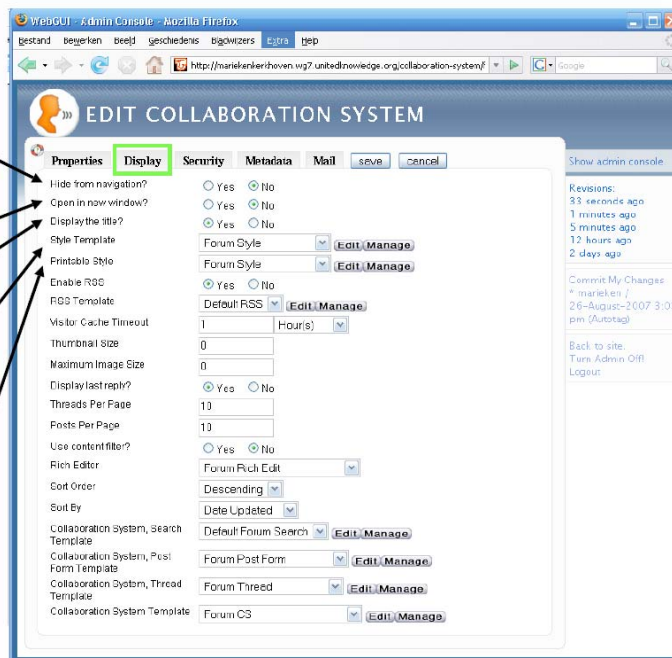
Open in new window: yes or no

Display title: <tmpl_if displayTitle>

Style template:

choose the style in witch your CS assets sets in

Printable Style: when you print your page



4/15/2009

donor.com community conference

110

And some new ones:

Enable RSS feed: yes or no

RSS template:

choose your template for RSS

Visitor cache timeout:

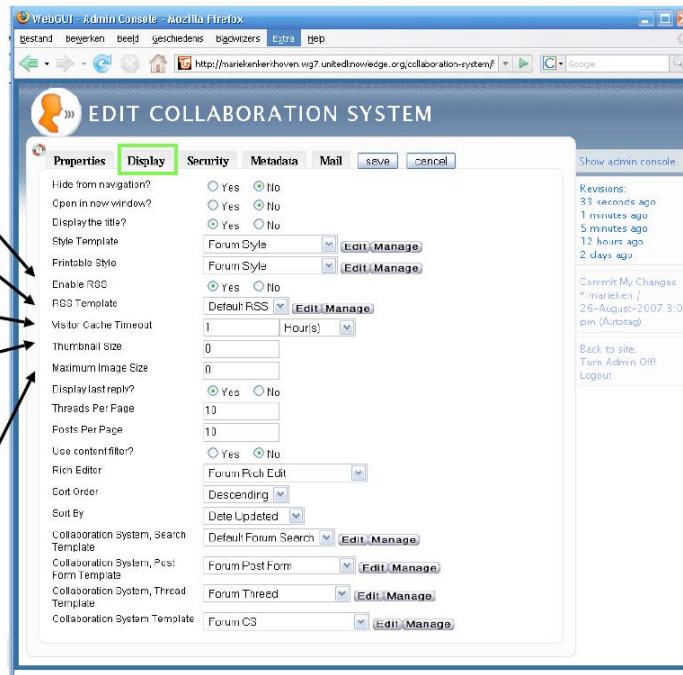
the time the cache is set for the visitor for this asset.

Thumbnail size:

size of thumb of image (avatar of photo) when loaded up in CS.
Can't be changed afterwards!

Max. image size:

size of uploaded photo. Can be changed after is uploaded.



4/15/2009

donor.com community conference

Display last reply:

in CS template:

`<tmpl_if displayLastReply>`

If set to yes, more variables become available, for example:

sortby.replies.url

lastReply.url

lastReply.title

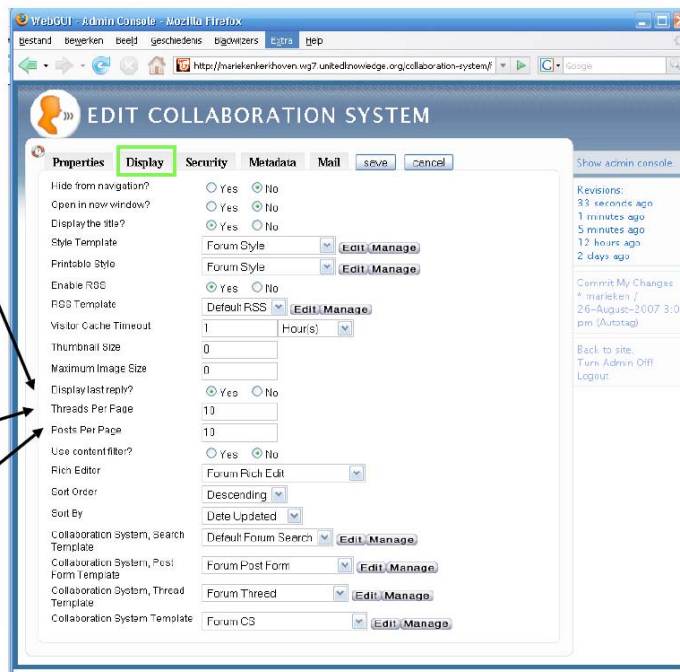
lastReply.username

Threads per page:

number of contributions in the thread view

Post per page:

number of contributions in the CS overview



4/15/2009

donor.com community conference

Use content filter:

to filter on specific words
like strong language
(waar instellen?)

Rich editor:

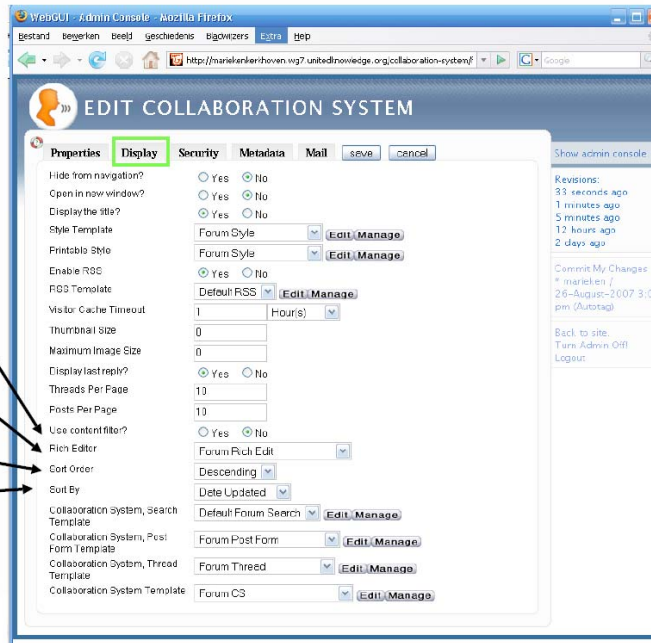
editor for editing text in a
text area (Forum Rich Edit or
Content Manager Rich Edit)

Sort order:

descending (up) or
ascending (down)

Sort by:

date, sequence, title, etc.



**And this we will go into in detail
later: the templates**

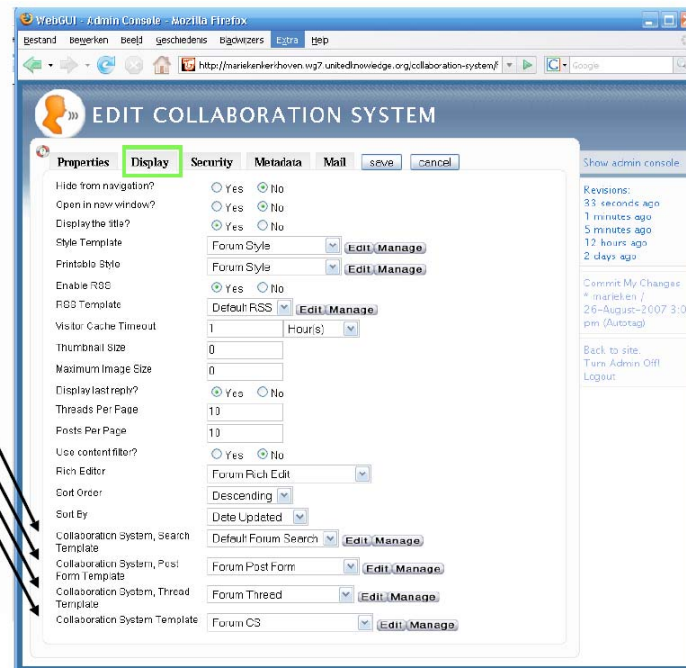
Select templates for:

Search

Post Form

Thread

Collaboration template (overview)



As in other assets

Encrypt content

Owner

Who can view:

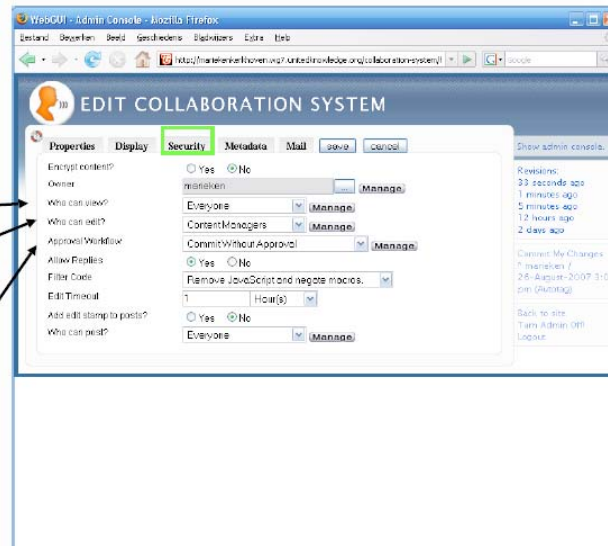
set to everyone or a specific group of people

Who can edit:

set to content managers or a specific group of people

Approval workflow:

when a post has to be approved before publishing



And this we will go into in detail later: the templates

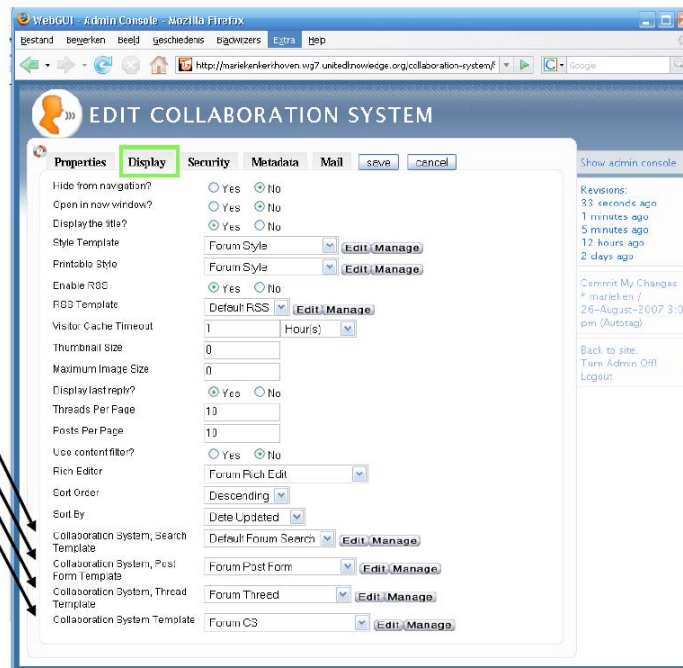
Select templates for:

Search

Post Form

Thread

Collaboration template (overview)



As in other assets

Encrypt content

Owner

Who can view:

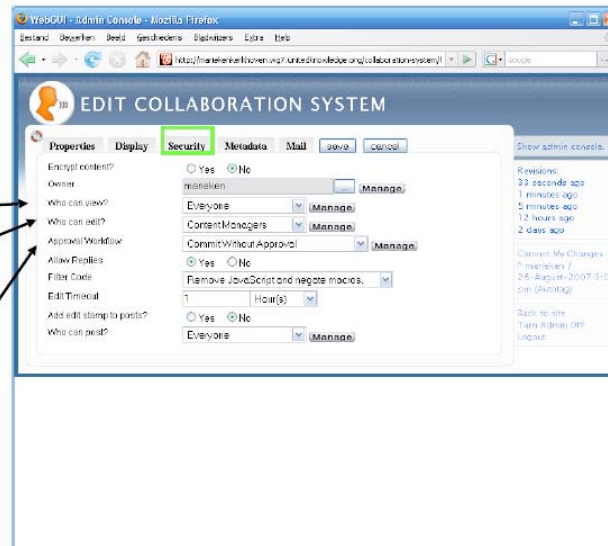
set to everyone or a specific group of people

Who can edit:

set to content managers or a specific group of people

Approval workflow:

when a post has to be approved before publishing



Allow replies:

can one reply on a post?

Filter code:

removes for example javascript or macros from a post

Edit timeout:

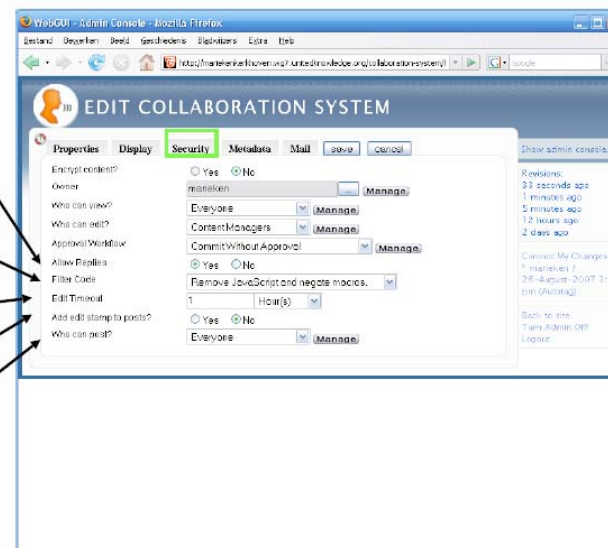
the time for a poster to edit his/her post

Add edit stamps: "Edited on <date>"

And last but not least:

Who can post?

Set to everyone or a specific group like Registered Users



As in other assets:

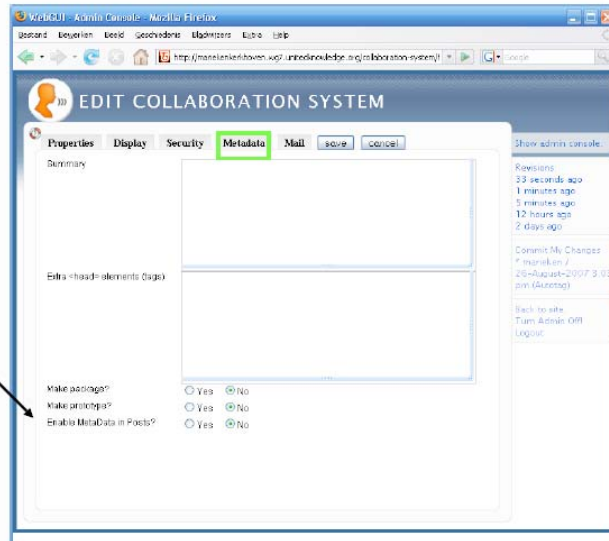
[Summary](#)

[Extra tags](#)

[Make package](#)

[Make prototype](#)

[Enable metadata in post](#): set to yes to enable metadata and to profile posts



It's possible to send and receive post by email. You need to have a dedicated mailbox.

Auto subscribe to thread: yes or no

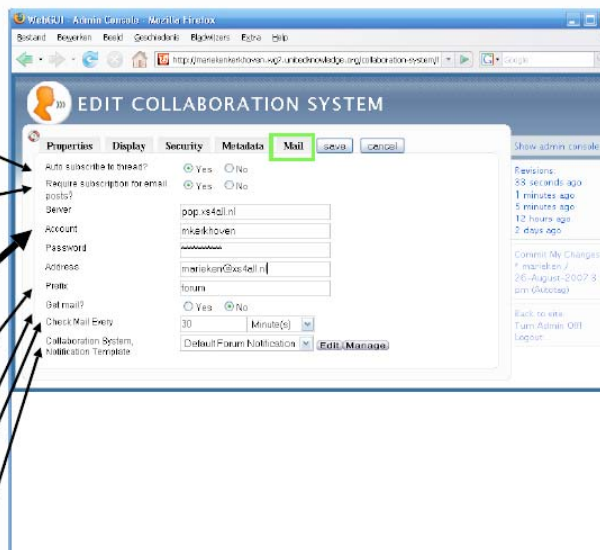
[Require subscription for email post?](#): do one need to be subscribed to be able to set post by email?

Fill in: [Server](#), [Account](#), [Password](#) and [address](#).

[Prefix](#): text in the subject field of the email

[Get mail](#): receive email when a new post is received in your forum

[Check mail](#): set time



Allow replies:

can one reply on a post?

Filter code:

removes for example javascript or macros from a post

Edit timeout:

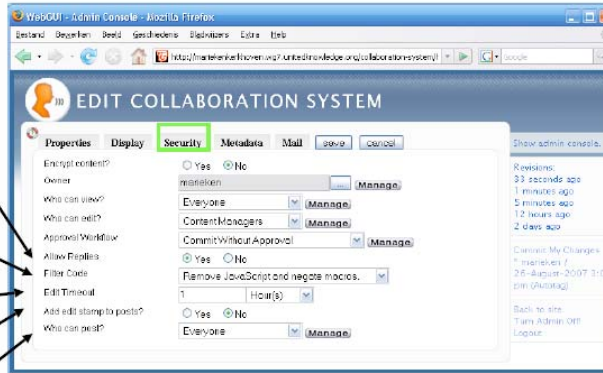
the time for a poster to edit his/her post

Add edit stamps: "Edited on <date>"

And last but not least:

Who can post?

Set to everyone or a specific group like Registered Users

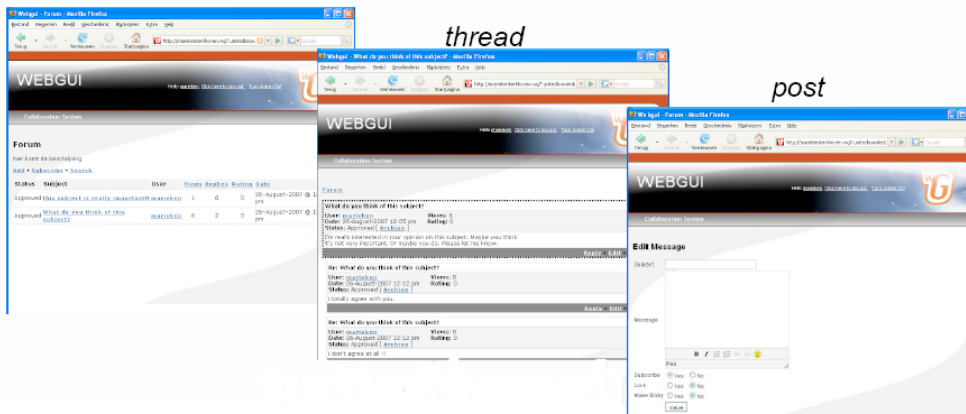


We will have a look at the 3 most important templates in detail:

CS overview

thread

post



CS Overview Template

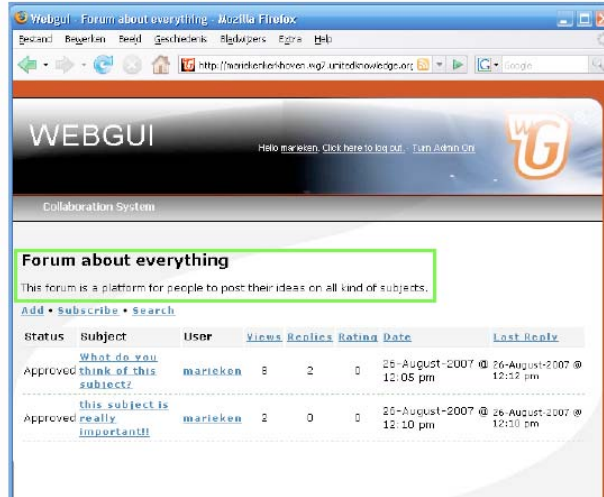
As always:

```
<a name="id<tmpl_var assetId>"
id="id<tmpl_var assetId>"></a>

<tmpl_if session.var.adminOn>
  <p><tmpl_var controls></p>
</tmpl_if>

<tmpl_if displayTitle>
  <h2> <tmpl_var title> </h2>
</tmpl_if>

<tmpl_if description>
  <tmpl_var description>
</tmpl_if>
```



4/15/2009

donor.com community conference

123

CS Overview Template

```
<tmpl_if user.canPost>
  <a href="<tmpl_var
  add.url>"><tmpl_var
  add.label></a>&#149;
</tmpl_if>

<tmpl_unless user.isVisitor>
  <tmpl_if user.isSubscribed>
    <a href="<tmpl_var
    unsubscribe.url>">
    <tmpl_var
    unsubscribe.label></
    a>
  </tmpl_if>
  <a href="<tmpl_var
  subscribe.url>"><tmpl_v
  ar subscribe.label></a>
</tmpl_if>&#149;
</tmpl_unless>

<a href="<tmpl_var
search.url>"><tmpl_var
search.label></a>
```



4/15/2009

donor.com community conference

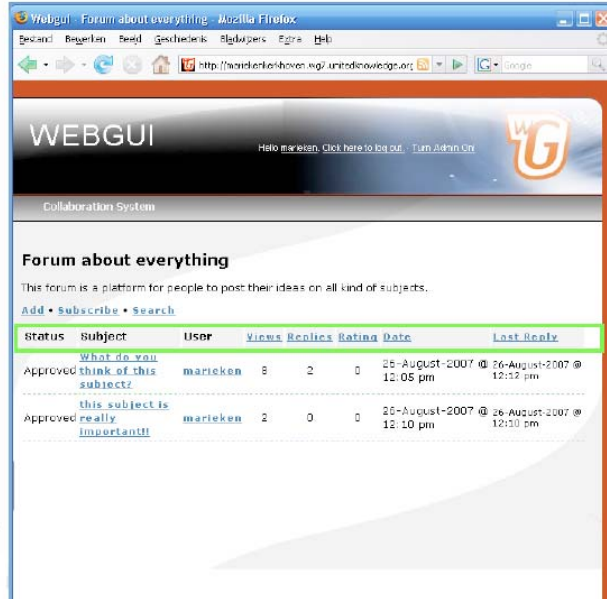
124

CS Overview Template

```

<table width="100%">
<tr>
  <tmpl_if user.isModerator>
    <td class="forumHead">
      <tmpl_var status.label> </td>
    </tmpl_if>
    <td class="forumHead"> <tmpl_var
      user.label> </td>
    <td class="forumHead"><a
      href="<tmpl_var sortby.views.url">">
      <tmpl_var views.label> </a></td>
    <td class="forumHead"><a
      href="<tmpl_var sortby.replies.url">">
      <tmpl_var replies.label> </a></td>
    <td class="forumHead"><a
      href="<tmpl_var sortby.rating.url">">
      <tmpl_var rating.label> </a></td>
    <td class="forumHead"><a
      href="<tmpl_var sortby.date.url">">
      <tmpl_var date.label> </a></td>
    <tmpl_if displayLastReply>
      <td class="forumHead"><a
      href="<tmpl_var sortby.lastreply.url">">
      <tmpl_var lastReply.label> </a></td>
    </tmpl_if>
  </tr>

```



4/15/2009

donor.com community conference

125

CS Overview Template

```

<tmpl_loop post_loop>
  <tr><tmpl_if user.isModerator>
    <td ><tmpl_var
      status></td>
    </tmpl_if>
    <td ><a href="<tmpl_var url">">
      <tmpl_var title></a></td>
    <tmpl_if user.isVisitor>
      <td ><tmpl_var username></td>
    <tmpl_else>
      <td ><a href="<tmpl_var
        userProfile.url">">
        <tmpl_var username></a></td>
    </tmpl_if>
    <td ><tmpl_var views></td>
    <td ><tmpl_var replies></td>
    <td ><tmpl_var rating></td>
    <td ><tmpl_var
      dateSubmitted.human> @
      <tmpl_var timeSubmitted.human></
      td>

```



4/15/2009

donor.com community conference

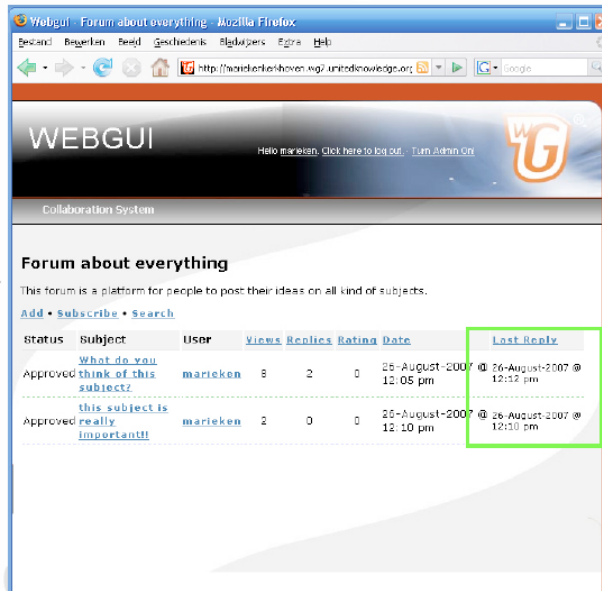
126

CS Overview Template

```

<tmpl_if displayLastReply>
  <td >
    <a href="<tmpl_var
lastReply.url">"><tmpl_var
lastReply.title></a>
    by
      <tmpl_if lastReply.user.isVisitor>
        <tmpl_var
lastReply.username>
      <tmpl_else>
        <a href="<tmpl_var
lastReply.userProfile.url">"><tmpl_var
lastReply.username></a>
      </tmpl_if>
    on <tmpl_var
lastReply.dateSubmitted.human> @
    <tmpl_var
lastReply.timeSubmitted.human>
  </td>
</tmpl_loop>
</table>

```



4/15/2009

donor.com community conference

127

CS Overview Template

```

<tmpl_loop post_loop>
  <tr><tmpl_if user.isModerator>
    <td ><tmpl_var
status></td>
  </tmpl_if>

  <td ><a href="<tmpl_var url">">
    <tmpl_var title></a></td>
  <tmpl_if user.isVisitor>
    <td ><tmpl_var username></td>
  <tmpl_else>
    <td ><a href="<tmpl_var
userProfile.url">">
    <tmpl_var username></a></td>
  </tmpl_if>

  <td ><tmpl_var views></td>
  <td ><tmpl_var replies></td>
  <td ><tmpl_var rating></td>
  <td ><tmpl_var
dateSubmitted.human> @
  <tmpl_var timeSubmitted.human></
td>

```



4/15/2009

donor.com community conference

128

Next: the thread template

a post
with its
replies



CS Thread Template

```

<div><a name="id<tmpl_var assetId>"
id="id<tmpl_var assetId>"></a></div>

<tmpl_if session.var.adminOn>
  <p><tmpl_var controls></p>
</tmpl_if>

<div style="float: left; width: 70%">
  <h2><a href="<tmpl_var collaboration.url>">
  <tmpl_var collaboration.title></a></h2>
</div>
  
```



CS Thread Template

```

<div style="width: 30%; float: left;
text-align: right;">
  <tmpl_if layout.isFlat>
    <a href="<tmpl_var layout.nested.url">">
      <tmpl_var layout.nested.label></a>
  <tmpl_else>
    <a href="<tmpl_var layout.flat.url">">
      <tmpl_var layout.flat.label></a>
  </tmpl_if>
</div>
<div style="clear: both;"></div>

<tmpl_loop post_loop>

  <tmpl_if layout.isNested>
    <div style="margin-left: <tmpl_var
depthX10>px;">
      </tmpl_if>

  <div class="postBorder<tmpl_if
isCurrent>Current</tmpl_if">

```



4/15/2009

donor.com community conference

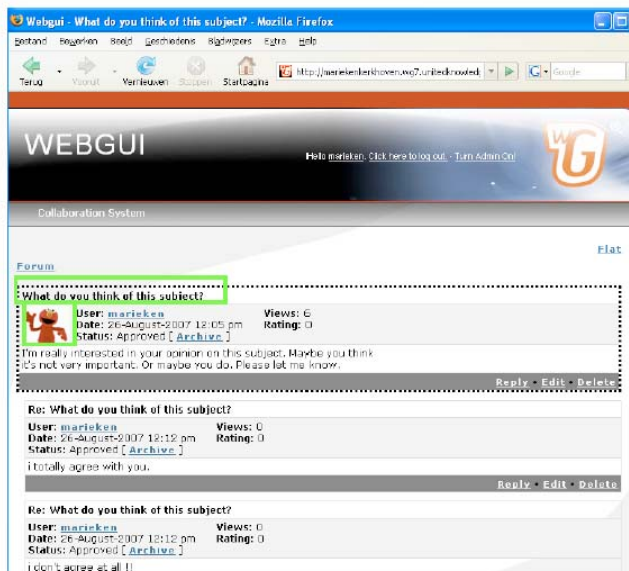
131

CS Thread Template

```

<tmpl_var title>
</div>
<div class="postData">
  <tmpl_if avatar.url>
    <div style="float: left; padding-right:
10px;">
      "
alt="<tmpl_var username">" />
    </div>
  </tmpl_if>
  <div style="float: left; padding-right:
25px;">

```



4/15/2009

donor.com community conference

132

CS Thread Template

```

<b><tmpl_var user.label>:</b>
<tmpl_if user.isVisitor>
  <tmpl_var username>
  <tmpl_else>
    <a href="<tmpl_var
userProfile.url"><tmpl_var
username></a>
  </tmpl_if>
<b><tmpl_var date.label>:</b>
<tmpl_var dateSubmitted.human>
<br />
</div>

```



4/15/2009

donor.com community conference

133

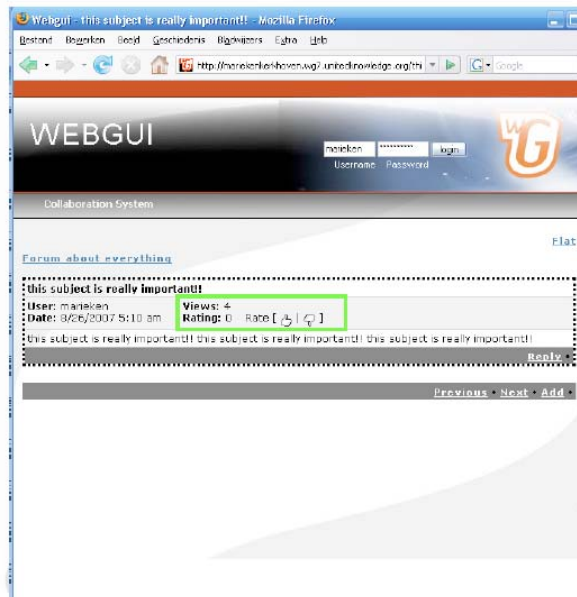
CS Thread Template

```

<div>
  <b><tmpl_var views.label>:</b>
  <tmpl_var views><br />
  <b><tmpl_var rating.label>:</b>
  <tmpl_var rating>

  <tmpl_if user.canPost>
    <tmpl_unless hasRated>
      <tmpl_var rate.label>
      [ <p style="display:inline;vertical-align:middle;"><a href="<tmpl_var
rate.url.thumbsUp"> alt="+"
style="border: 0px;vertical-align:middle;" /
></a></p> | <p
style="display:inline;vertical-align:middle;"><a href="<tmpl_var
rate.url.thumbsDown">
alt="-" style="border: 0px;vertical-align:middle;" /></a></p> ]
    </tmpl_unless>
  </tmpl_if>

```



4/15/2009

donor.com community conference

134

CS Thread Template

```

<div class="postMessage">
  <tmpl_var content>

<tmpl_loop attachment_loop>
  <div style="float: left; padding: 5px;">
    <a href="<tmpl_var url">">
      <tmpl_if isImage>
        " />
      <tmpl_else>
        " /
></a> <a href="<tmpl_var
url">"><tmpl_var filename>
      </tmpl_if>
    </div>
  </tmpl_loop>
<div style="clear: both;"></div>

```



4/15/2009

donor.com community conference

135

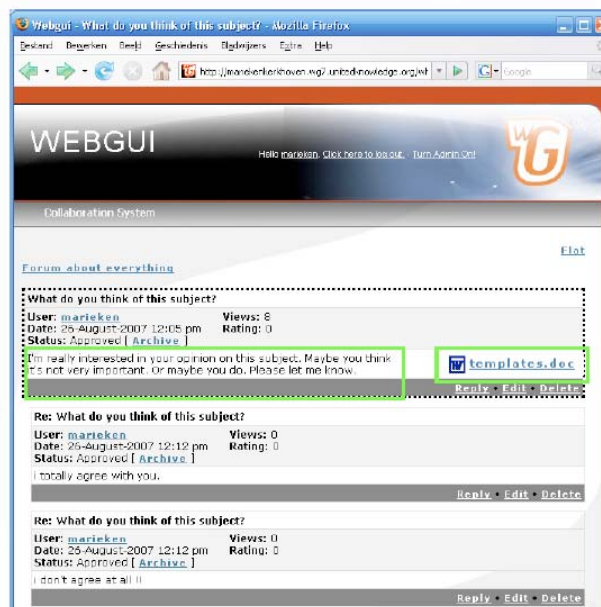
CS Thread Template

```

<div class="postMessage">
  <tmpl_var content>

<tmpl_loop attachment_loop>
  <div style="float: left; padding: 5px;">
    <a href="<tmpl_var url">">
      <tmpl_if isImage>
        " />
      <tmpl_else>
        " /
></a> <a href="<tmpl_var
url">"><tmpl_var filename>
      </tmpl_if>
    </div>
  </tmpl_loop>
<div style="clear: both;"></div>

```



4/15/2009

donor.com community conference

136

CS Thread Template

```

<tmpl_unless isLocked>
  <div class="postControls">
    <tmpl_if user.canReply>
      <a href="<tmpl_var
reply.url">"><tmpl_var reply.label></a>
&bull;
    </tmpl_if>
    <tmpl_if user.canEdit>
      <a href="<tmpl_var
edit.url">"><tmpl_var edit.label></a>
&bull;
      <a href="<tmpl_var
delete.url">"><tmpl_var delete.label></a>
    </tmpl_if>
  </div>
</tmpl_unless>
</div>
<tmpl_if layout.isNested></div></tmpl_if>
</tmpl_loop>

```



4/15/2009

donor.com community conference

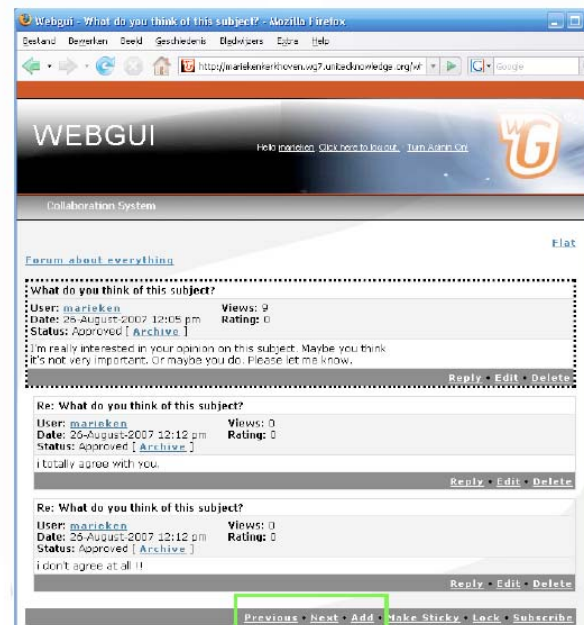
137

CS Thread Template

```

<div style="margin-top: 20px;"
class="bottomControls">
<tmpl_if previous.url>
  <a href="<tmpl_var
previous.url">"><tmpl_var
previous.label></a> &bull;
</tmpl_if>
<tmpl_if next.url>
  <a href="<tmpl_var
next.url">"><tmpl_var next.label></a>
&bull;
</tmpl_if>
<tmpl_if user.canPost>
  <a href="<tmpl_var
add.url">"><tmpl_var add.label></a>
&bull;
</tmpl_if>

```



4/15/2009

donor.com community conference

138

CS Thread Template

```

<tmpl_if user.isModerator>
<tmpl_if isSticky>
  <a href="<tmpl_var unstick.url">">
  <tmpl_var unstick.label></a> <tmpl_else>
  <a href="<tmpl_var stick.url">"><tmpl_var
  stick.label></a> </tmpl_if>
<tmpl_if isLocked>
  <a href="<tmpl_var unlock.url">">
  <tmpl_var unlock.label></a> <tmpl_else>
  <a href="<tmpl_var lock.url">"><tmpl_var
  lock.label></a>
</tmpl_if> </tmpl_if>
<tmpl_unless user.isVisitor>
  <tmpl_if user.isSubscribed>
  <a href="<tmpl_var unsubscribe.url">">
  <tmpl_var unsubscribe.label></a>
  <tmpl_else>
  <a href="<tmpl_var subscribe.url">">
  <tmpl_var subscribe.label></a>
  </tmpl_if>
</tmpl_unless>
</div>

```



4/15/2009

donor.com community conference

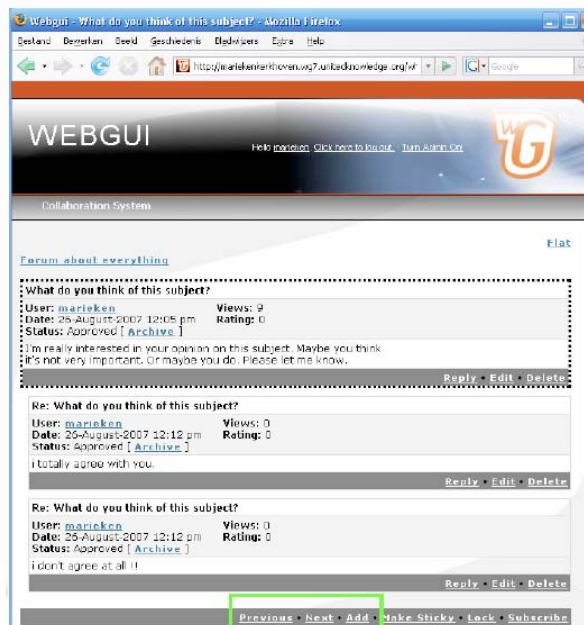
139

CS Thread Template

```

<div style="margin-top: 20px;"
class="bottomControls">
<tmpl_if previous.url>
  <a href="<tmpl_var
  previous.url">"><tmpl_var
  previous.label></a> &bull;
</tmpl_if>
<tmpl_if next.url>
  <a href="<tmpl_var
  next.url">"><tmpl_var next.label></a>
  &bull;
</tmpl_if>
<tmpl_if user.canPost>
  <a href="<tmpl_var
  add.url">"><tmpl_var add.label></a>
  &bull;
</tmpl_if>

```



4/15/2009

donor.com community conference

140

CS Post Template

And then: the post template

The form with which a new contribution is posted or an existing post is edited.



CS Post Template

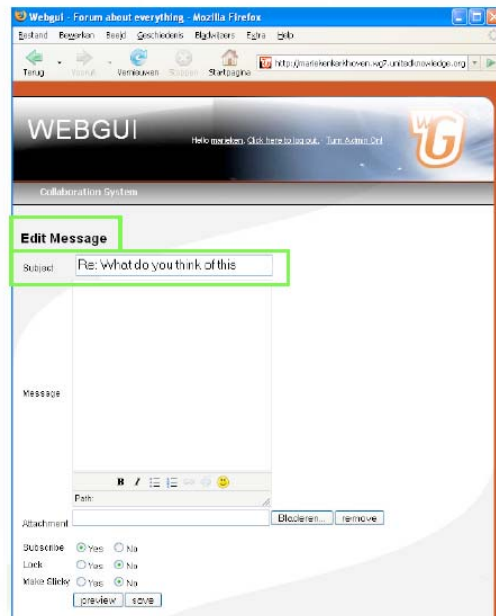
```

<a name="id<tmpl_var assetId>"
  id="id<tmpl_var assetId>"></a>

<tmpl_if preview.title>
  <p><b><tmpl_var preview.title></b></p>
</tmpl_if>

<tmpl_if preview.content>
  <p><tmpl_var preview.content></p>
</tmpl_if>

<h2><tmpl_var message.header.label></h2>
  
```



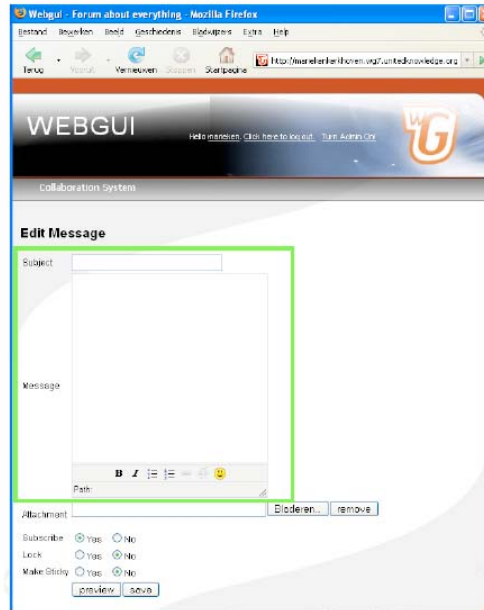
CS Post Template

```

<tmpl_var form.header>

<table class="defaultPostForm">
  <tmpl_if isNewPost>
    <tmpl_if user.isVisitor>
      <tr>
        <td><tmpl_var
visitorName.label></td>
        <td><tmpl_var
visitorName.form</td>
      </tr>
    </tmpl_if>
  </tmpl_if>
  <tr>
    <td><tmpl_var subject.label></td>
    <td><tmpl_var title.form></td>
  </tr>
  <tr>
    <td><tmpl_var message.label></td>
    <td><tmpl_var content.form></td>
  </tr>
</table>

```



4/15/2009

donor.com community conference

143

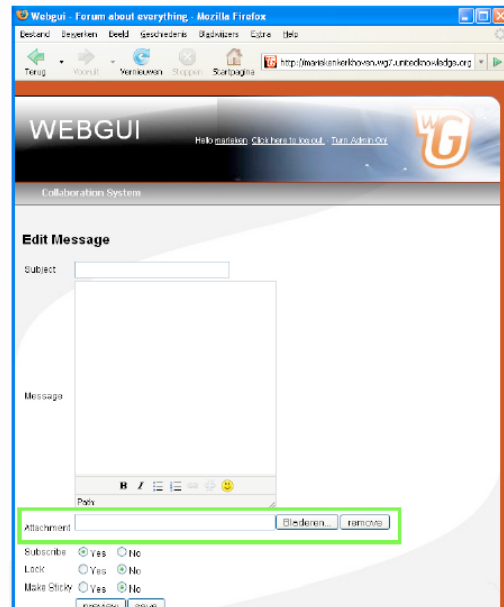
CS Post Template

```

<tmpl_unless isReply>
  <tmpl_loop meta_loop>
    <tr>
      <td><tmpl_var name></td>
      <td><tmpl_var field></td>
    </tr>
  </tmpl_loop>
</tmpl_unless>

<tmpl_if attachment.form>
  <tr>
    <td><tmpl_var attachment.label></
td>
    <td><tmpl_var
attachment.form</td>
  </tr>
</tmpl_if>

```



4/15/2009

donor.com community conference

144

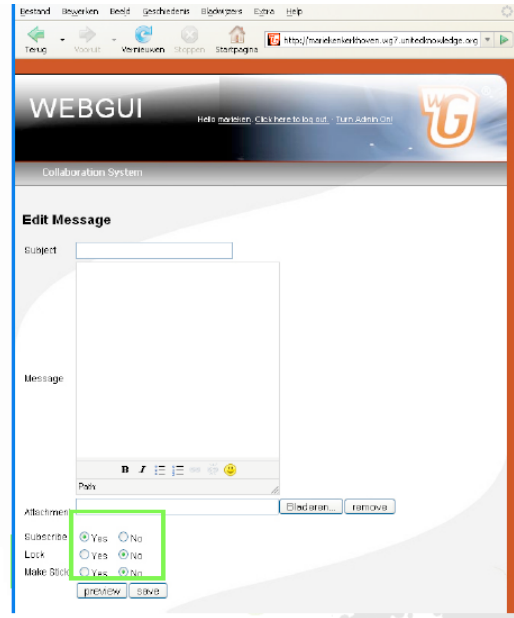
CS Post Template

```

<tmpl_if isNewPost>
<tmpl_unless user.isVisitor>
  <tr>
    <td><tmpl_var subscribe.label></td>
    <td><tmpl_var subscribe.form></td>
  </tr>
</tmpl_unless>

<tmpl_if isNewThread>
  <tmpl_if user.isModerator>
  <tr>
<td><tmpl_var lock.label></td>
    <td><tmpl_var lock.form></td>
  </tr>
  <tr>
    <td><tmpl_var stick.label></td>
    <td><tmpl_var sticky.form></td>
  </tr>
</tmpl_if>
</tmpl_if>

```



4/15/2009

donor.com community conference

145

CS Post Template

```

<tr>
  <td></td>

<td><tmpl_if usePreview><tmpl_var
form.preview></tmpl_if>

<tmpl_var form.submit></td>
</tr>
</table>

<tmpl_var form.footer>

<tmpl_if isReply>
  <p><b><tmpl_var reply.title></b></p>
  <tmpl_var reply.content>
</tmpl_if>

```



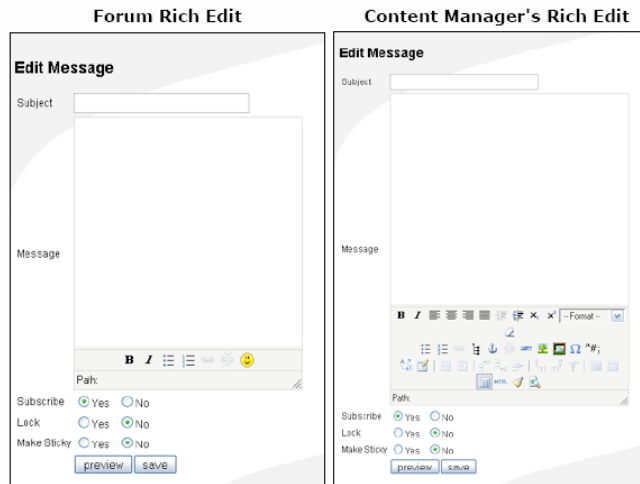
4/15/2009

donor.com community conference

146

CS Post Template

Two kind of editors
you set them in
settings under
display



4/15/2009

donor.com community conference

CS Post Template

Variables in Post template used in
different post form

- post_loop
- title
- username
- userProfile.url
- dateSubmitted.human
- views
- rating
- user.isPoster
- status
- content
- edit.url
- userDefined1
- userDefined2
- userDefined3
- userDefined4
- userDefined5



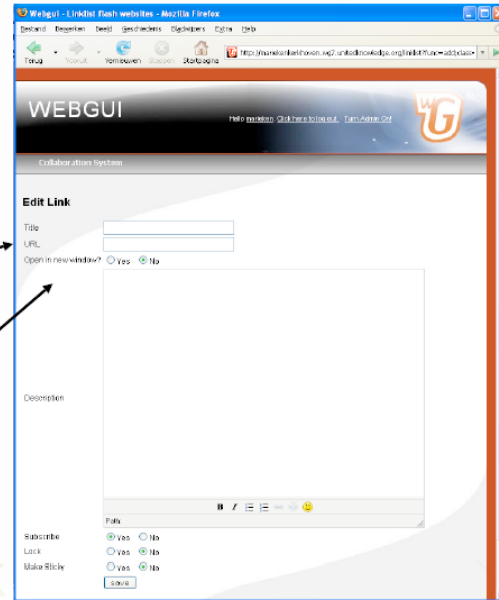
4/15/2009

donor.com community conference

CS Post Template

Use of `userDefined1` / `userDefined5`
 For example: in the Link List Submission Form

```
[...] <tr>
<td><tmpl_var title.label></td>
<td><tmpl_var title.form></td>
</tr>
<tr>
<td><tmpl_var url.label></td>
<td><tmpl_var userDefined1.form></td>
</tr>
<tr>
<td><tmpl_var newWindow.label></td>
<td><tmpl_var userDefined2.form.yesNo></td>
</tr>
<tr>
<td><tmpl_var description.label></td>
<td><tmpl_var content.form></td>
</tr> [...]
```



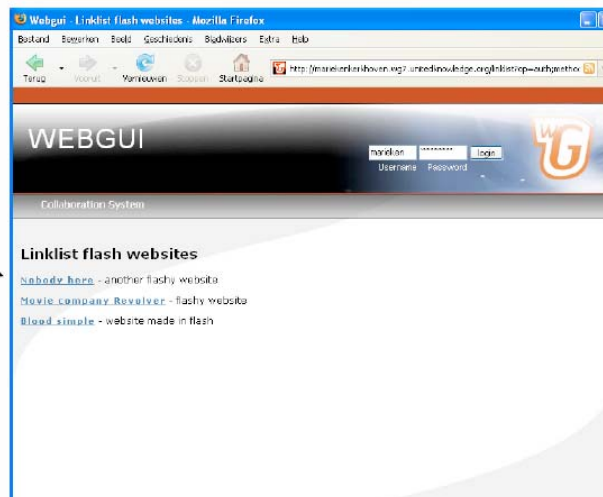
CS Post Template

Use of `userDefined1` / `userDefined5`

they will show up in you CS of thread view as in de link list

URL

And also open new window is true.



CONCLUSION

- ✓ Collaboration System has different views
- ✓ It has many different default templates
- ✓ It has many different settings
- ✓ It has many variables
- ✓ You can change the default templates to your liking
- ✓ By combining the settings, variables and templates you can make a CS the way you want.

Supplement 1: CMS Template Post Variables

Formname	Template variable	default post	default submission	FAQ	Job	Link list	Request Tracker
visitorName label:	visitorName Visitor Name	visitorName.form visitorName.label	visitorName.form visitorName.label	no no	no no	no no	visitorName.form visitorName.label
title label:	title Subject/Title/Question/Job Title	title.form subject.label	title.form title.label	title.form,textarea question.label	title.form job.title.label	title.form title.label	title.form subject.label
synopsis label:	synopsis Summary	no	synopsis.form synopsis.label	no no	no no	no no	no
content label:	content Message/Body/(Job) Description	content.form message.label	content.form body.label	content.form answer.label	content.form job.description.label	content.form description.label	content.form description.label
userDefined1 label:	userDefined1 Compensation/URL	no	no	no	userDefined1.form compensation.label	userDefined1.form uri.label	custom severity.label
userDefined2 label:	userDefined2 Location	no	no	no	userDefined2.form location.label	userDefined2.form, yes newWindow.label	no
userDefined3 label:	userDefined3 Job Requirements	no	no	no	userDefined3.form,htmlarea job.requirements.label	no	no
userDefined4 label:	userDefined4	no	no	no	no	no	no
userDefined5 label:	userDefined5	no	no	no	no	no	no
storageId_action label:	attachment_loop Attachment	attachment.form attachment.label	attachment.form attachment.label	attachment.form attachment.label	attachment.form attachment.label	attachment.form attachment.label	attachment.form attachment.label
subscribe label:	user.isSubscribed Subscribe	subscribe.form subscribe.label	subscribe.form subscribe.label	subscribe.form subscribe.label	subscribe.form subscribe.label	subscribe.form subscribe.label	subscribe.form subscribe.label
lock label:	isLocked Lock	lock.form lock.label	lock.form lock.label	lock.form lock.label	lock.form lock.label	lock.form lock.label	lock.form lock.label
stick label:	isSticky Make Sticky	sticky.form stick.label	sticky.form stick.label	sticky.form stick.label	sticky.form stick.label	sticky.form stick.label	sticky.form stick.label
	preview.title preview.content						

Supplement 1: CMS Template Post Variables

usual variables	cs variables	cs template specific variables	cs specific post variables	post common variables
<ul style="list-style-type: none"> assetId session.var.adminOn control displayTitle title description user.isVisitor menuItem url 	<ul style="list-style-type: none"> user.canPost add.uri user.isSubscribed unsubscribe.uri unsubscribe.label subscribe.uri subscribe.label user.isModerator 	<ul style="list-style-type: none"> thumbnail displayLastReply search.uri sortBy.views.uri sortBy.replies.uri sortBy.rating.uri sortBy.date.uri sortBy.lastreply.uri sortBy.karmaRank.uri 	<ul style="list-style-type: none"> url thumbnail synopsis dateSubmitted.human timeSubmitted.human reply displayLastReply dateUpdated.human lastReply.uri lastReply.title lastReply.user.isVisitor lastReply.username lastReply.userProfile.uri lastReply.username lastReply.dateSubmitted.human lastReply.timeSubmitted.human 	<ul style="list-style-type: none"> post_loop title username userProfile.uri dateSubmitted.human views rating user.isPoster status content edit.uri userDefined1 userDefined2 userDefined3 userDefined4 userDefined5
usual variables	cs variables	thread specific variables	thread specific post variables	post common variables
<ul style="list-style-type: none"> assetId session.var.adminOn control displayTitle title description user.isVisitor menuItem url 	<ul style="list-style-type: none"> user.canPost add.uri user.isSubscribed unsubscribe.uri unsubscribe.label subscribe.uri subscribe.label user.isModerator 	<ul style="list-style-type: none"> replies.Allowed replies.label collaboration.uri collaboration.title layout.nested.uri layout.nested.label layout.flat.uri layout.flat.label layout.isNested layout.isNested mpi_var_depth_X10 karmaEnabled karmaRank transfer.karma.form 	<ul style="list-style-type: none"> attachment_loop url isImage thumbnail filename icon isThreadRoot avatar.uri hasRated rate.uri thumbsUp thumbsUp.icon.uri rate.uri thumbsDown thumbsDown.icon.uri isArchived unarchive.uri archive.uri isLocked user.canReply reply.uri delete.uri previous.uri next.uri isCurrent isSticky unstick.uri stick.uri unlock.uri lock.uri 	<ul style="list-style-type: none"> post_loop title username userProfile.uri dateSubmitted.human views rating user.isPoster status content edit.uri userDefined1 userDefined2 userDefined3 userDefined4 userDefined5

BREAK

I. The DWAPI Overview (Mike)

- Overview of an API
- The webware DWAPI overview and life-cycle
- From SOAP to the DWAPI

II. Building blocks of the DWAPI

- Connection
- Behavior
- Presentation
- Webware Example Templates (identify & multi-method)
- syncProfile

III. Webware Goodies

- API = Application Programming Interface
- WebWare is one of many DonorWare APIs
- WebWare is the “donor self-service” API
- WebWare API Reference:

donor.com/ref/donorware/app/webware

- **Module: DonorWare::App::FetchCodes**
- **Method: fetch_codes()**
- **Parameters: table => 'dept'**

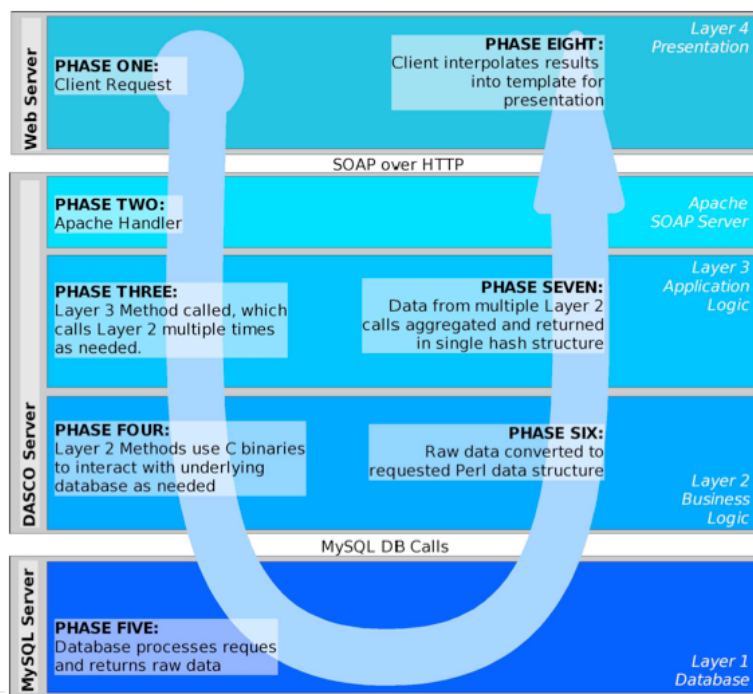
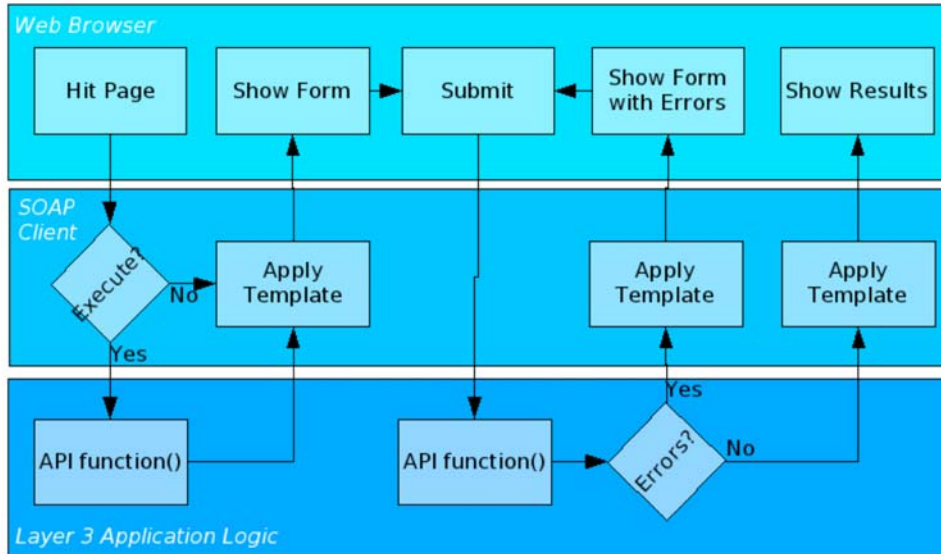
<http://donor.com/ref>

- donor.com APIs are wrappers around DASCO binaries

```
$ dwcodes dept  
123|Dept 123  
456|Dept 456
```

- WSClient Asset/Wobject
- DWClient Asset/Wobject
- Client calls WebWare API
- Client template shows results or errors

These four bullets will be covered in greater detail later.



WSClient

- XML
- No encryption or SSL
- No compression
- HTTP envelope

2-3 seconds overhead per request

DWClient

- Storable
- Symmetric encryption
- Zlib compression
- Raw TCP

< 1 second overhead per request

236 ms - binary getname call to get the data

86 ms - dwapid overhead for serialization/encryption, etc

====

322 ms - time for API to return to socket

80 ms - Internet for Calgary to Denver

====

402 ms - packet arrives at client

151 ms- time for decryption/deserialization and population of gui

====

553 ms

Before: SyncProfilesFromDASCO

- Hourly batch job

After: dwpush

- new getchglog binary in DASCO
- triggers DASCO server event
- HTTP Request to WebGUI7 Workflow
- Changes reflected in 30 seconds or less

Part II

The building blocks of the DWAPI

To best understand the basic building blocks of the DWAPI methodology, think of these three components

- I. **API Connection** – the DASCOServer it connects to and the API Module (layer 3) used
- II. **API Behavior** – What and how data is passed through API
- III. **API Presentation** – How the data is displayed to the end user/ How the end-user interacts with the data

A VISUAL REPRESENTATION OF THE THREE COMPONENTS

- ✓ The Connection
- ✓ The Behavior
- ✓ The Presentation



The config DB is another component of behavior.

This is currently not configurable by clients, but will be sometime in later 08.

We will briefly touch on the role of the config DB.

The Connection

DWAPI server (host:port)	test2.api.dwvpn.net:9002
DWAPI module	DonorWare::App::WebWare
DWAPI Method/Call	multi_method

The connection is established by three fields in the DW Client

DWAPI Server tells the API which DASCO server the API connects to and which port to use. Your client server will be "XXX.api.dwvpn.net:9002", where XXX is your client abbreviation (GFA, FH, MAF, MMI, etc). Client's always use port 9002

DWAPI Module is the file that contains the methods

A high-level (layer 3 in DonorWare's 4-tier architecture) API that allows either donor.com's presentation layers (layer 4) to interact with donor-facing functionality, such as address changes, donations, giving history, etc. The WebWare API in turn calls the layer 2 dascoXFS business logic API, which in turn interacts with the actual underlying data, while enforcing the security and business rules of the organization.

- ✓ Find a list of all webware methods at <http://donor.com/ref/donorware/app/webware>
- ✓ The 2 pertaining to your website are app::webware & app::webware::sponsorship

STOP – Let's look at the webware API/module and all it's methods

D

**W
A**

PI Method/Call (a very specific function call to DASCO where it sends and retrieves information)

The Behavior Part A – Config DB

Helps setup the basic behavior of how webware module interacts with DASCO

- ✓ Currently do not have access to configure, but we can let them see what's configured.
- ✓ Allows you to set standard variables that can be used to change the behavior of how an API works
 - ✓ Default designations
 - ✓ Motivations/Transformations
 - ✓ Drop-down lists
 - ✓ Change range of designations allowable
 - ✓ Change notification emails
 - ✓ Drive template used for auto-emails (like after donation)

- ✓ WebWare APIs can behave differently for different clients.
- ✓ Each client can tweak the business logic of the WebWare API via the ConfigDB
- ✓ Currently requires DonorWare Support
- ✓ We plan to move into main DASCO code maintenance for each client this coming year

STOP – Let's briefly look at the "test2" config db settings. Our conference site is running off of the test2 dasco.

- dateSlot
- desigByFreqAndAmt
- dynamicMotivation
- motivPrefix
- prefixLtdLen
- rangeTable
- regexList
- regexTable
- replaceIf
- replaceInvalidDesig
- replaceInvalidMotiv
- stringTable
- stripSpace
- upperCase

Wwym-ABCD becomes W084-ABCD

This allows generating a whole group of motivations from a template, which motivations contain codes for year and month. For example, motivation template Wcym-MOTV would become W08C-MOTV in 2009-12 and W081-MOTV in 2009-01.

```
transformations
0 motiv
```

```
transform_motiv
0 xfMotvStripSpace
1 xfMotvDynamic
2 xfMotvUpperCase
```

```
xfMotvStripSpace
transform => stripSpace
input => motiv
```

```
xfMotvDynamic
transform => dynamicMotivation
nature => data
input => prev
```

```
xfMotvUpperCase
transform => upperCase
input => prev
```

If motivation is invalid (or inactive), change it to 'FALL-BACK'.

```
transformations
  0  motiv

transform_motiv
  0  xfMotvFallback

xfMotvFallback
  transform => replaceInvalidMotiv
  input => motiv
  fallback => 'FALL-BACK'
```

Designation to be 8000 for onetime gifts; else, with monthly RCC, 8001 for gifts <= \$30 but 8002 for gifts > \$30.

Set the designation to 8001 initially (per method arguments), then conditionally replace it with one of the other values.

```
transformations
  0  desig
```

```
transform_desig
  0  xfDesgByAmt
  1  xfDesgByFreq
```

```
xfDesgByAmt
  transform => replacelf
  nature => data
  input => desig
  replace => 8002
  cvar => amt
  cmp => '>'
  cval => 30
```

```
xfDesgByFreq
  transform => replacelf
  nature => data
  input => prev
  replace => 8000
  cvar => freq
  cmp => 'eq'
  cval => 'onetime'
```

The Behavior Part B – DWAPI Call Parameters

DWAPI Method Parameters help control the specific behavior for each module

- ✓ Parameters (required)]
- ✓ Parameters (standard)]
- * See documentation for list

DWAPI Call Parameters

```
'SESSION' => '^var("sessionId");',  
'GOMOTIV' => '^FormParam("goMotiv");',  
'BROWSER' => '^Env("HTTP_USER_AGENT");',  
'URL' => '^Env("REQUEST_URI");',  
AUTH_TOKEN => '^AuthToken();',  
ID_CODE => '^var("userId");',  
method0 => 'clear_session',  
method1 => 'identify',  
method2 => 'donation', ...
```

Important Notes

<http://donor.com/ref/donorware/app/webware>

- ✓ Your default DWAPI clients have the standard/common parameters.
- ✓ All new donor.com sites now have an archive of the standard DWclients and their corresponding templates.
- ✓ When your site is moved to webgui 7.4 you can request a package of all the default clients and templates as a reference.

Macros Used in the DWAPI call params

- ^FormParam – Allows you to pass the parameter via the url
- ^var – used for 'session' to pull in the sessionId variable
- ^Env – can be used to pass a web server environment variable

DWAPI Webware Templates

Webware DWAPI Templates

✓ Just like WebGUI templates, webware DWAPI templates use template the `html::template` language.

- ✓ There are 4 key elements to each template
- API Transport Errors
 - Prepopulation Mode
 - Method Errors
 - Successful Results

Three `HTML::Template` Building Blocks:

`TMPL_VAR`

`TMPL_LOOP`

`TMPL_IF/TMPL_ELSE/TMPL_UNLESS`

IDENTIFY

Identify is tight-rope between security and usability

You want to make sure the donor has to put in enough information to be uniquely identify without making it difficult.

Some clients seeks a “donate and go” (no dup check) while others want to check unique components against the DASCO (dup check), requiring a greater amount of “being identified” against DASCO.

To find out more about identify and it's role in security and checking for duplicate accounts -

https://donor.com/manuals_webware2_dupcheck

Let's look at the identify template

4/15/2009

donor.com community conference

179

Identify syncProfile (a very very “layman's” explanation)

1. First account is created in DASCO (but not in webGUI)
2. syncProfile takes the account from DASCO (DW PUSH) and “pushes” it to webgui to create a WebGUI account.
3. When a user updates their account preferences (address, phone, email, etc), syncPROFILE has the role of making sure that the webgui account and DASCO account data are synced.

4/15/2009

donor.com community conference

180

syncProfileFromDasco

- Runs “live” via DWPUSH
- Keeps name, email, etc in sync
- DASCO Special Lists can drive WebGUI groups to control site access
- Uses WebGUI workflow for realtime DASCO syncs
- Uses a Custom Output Format, so you CAN include other fields
- You could then use those other fields to drive dynamic personalization in WebGUI

For about 95% of donor.com clients, multi-method is used as the primary donation page.

Multi-method accomplishes “identify” and takes the donation at the same time.

- ✓ Prompts existing online donors to login.
- ✓ Prompts non-online DASCO accounts to establish a WebGUI account that is synced with their DASCO account
- ✓ Creates new accounts for first time donors

- allows the aggregation of multiple API methods into one call
- Uses one big template with template info for each API method called
- Uses DynamicParams macro to gather all the parameters

```
'SESSION' => '^var("sessionId");',
'GOMOTIV' => '^FormParam("goMotiv");',
AUTH_TOKEN => '^AuthToken();',
ID_CODE => '^var("userId");',
method0 => 'clear_session',
method1 => 'identify',
method2 => 'donation',
'clear_session_clear_desig' =>
  '^FormParam("clr_dsg");',
'identify_IP_ADDRESS' => '^Env("REMOTE_ADDR");',
'identify_BROWSER' => '^Env("HTTP_USER_AGENT");',
'identify_SKIPSTAGE3' => '1',
'identify_REFERER' => '^Env("HTTP_REFERER");',
'clear_session_clear' => '^FormParam("clr_ssn");',
'DONATION_DONMOTIV' => '^FormParam("IDENTIFY_MOTIV");',
'DONATION_DESIG' => '^FormParam("DESIG");',
^DynamicParam('IDENTIFY_*');,
^DynamicParam('DONATION_*');
```

Multi Method Examples

Simple Donation

- Identify
- Donate

Simple Checkout

- Identify
- Checkout

Complex

- Identify
- Form Handler
- Donation