Twingle for Developers

A Technical Introduction

Quick Look


Twingle
New
Name
Local Content
Inbox
Welco...
New D...
Drafts
Articles
Notes
Events
People
oscom.org
OSCOM ...
Conferen...
CMS Matrix
Projects
Mailing Li...
Blog
Organiza...
Welcome
Welcome to Twingle
save
Contents
Source
Preview
Properties
Web Manage
Welcome to Twingle
Welcome to Twingle
Your Twingle is now setup and running.

What is Twingle?

- Unsolved problems
  - Usability sucks for all content management
  - We dilute scarce talent across projects
- Twingle is an OSCOM project to improve the authoring experience for WCM

Mission

- Significantly improve L/C/C
  - *Locate*: How do I find the resources to work with during authoring?
  - *Create*: Can I quickly and conveniently add content, while obeying "No surprises, No alarms?"
  - *Collaborate*: How do I work with others?

Audience

- Content authors
  - Secretaries, producers, and smart people who are in a hurry
- Non-audience
  - Developers, designers, sysadmins
  - Twingle != (Emacs, Visual Studio. Dreameaver, or MMC)

Focus

- Improving content management usability
- Engage the implementors and solve shared problems
- Push the mission and brand of OSCOM

Philosophy

- We can best help ourselves by working together
- Don’t invent new standards
- Leaving stuff out is more important than putting stuff in
- Most of the work is in “finishing”

Architecture

- True "MVC" with local state
- Server sends RDF "model" of *all content*
- JavaScript "controller" mediates local/ remote models (datasources & DAV)
- XUL, CSS, and specifically XBL provide a widget-oriented "view"

Technology

- XUL (chrome, XBL)
- JavaScript
- CSS
- RDF
- XML
- HTTP/DAV

Server Support

- Discoverable configuration
- contents.rdf
- HTTP PUT for creation/updating
- HTTP DELETE for deletion

Waahhh!

- “JavaScript is for l00zers!” *Bzzt.*
- “Mozilla is slow and ugly!” *Bzzt.*
- “Thick clients are so 90’s!” *Bzzt.*
- “RDF is semantic hogwash!” *Bzzt.*
- “DAV chafes my butt!” *Double Bzzt.*

In Action

Sprint Goals

- Make the “go/no go” decision
- Get to x-server DnD
- Orientation for new twinglers
- Establish patters for running project
- Make decisions on philosophy and architecture
- Prepare for OSCOM 3

Getting Started

- Follow instructions in *doc* dir
- Install jslib from jslib.mozdev.org
- Users or evaluators install an XPI
- Developers:
  - Add lines to `$MOZ/chrome/installed-chrome.txt`
  - Symlink to twingle/twingle

Mozilla Setup

- These are covered in debugging.txt
- Turn on pref for dump()
- Turn on “Disable XUL Cache”
- In Venkman (JS Debugger), unselect “Exclude chrome files”

Launch Pattern

- Read doc/debugging.txt
- From console/shell, run `$MOZ/mozilla-bin -chrome://twingle/content -jsconsole` and watch debug output
- Even better:
  - Use DOM Inspector
  - Use Venkman (JS Debugger)

UI Structure

- Main Twingle window has four regions
  - Menu
  - Toolbar (perhaps deprecated)
  - Navigation
  - Workspace (perhaps later with toolbox)

Directory Structure

- Top dirs mimic all Moz chrome apps
- twingle/content, twingle/skins, twingle/locale
- twingle/content mimics UI Structure
  - content/menu, content/toolbar
  - content/navigation, content/workspace

Overlay Structure

- Entry point of Twingle is content/twingle.xul
- The <window> element calls twingle.js to initialize
- Contains only “overlays”
- Thus, gets the Twingle UI from other files
  - menu/menuoverlay.xul
  - toolbar/toolbaroverlay.xul etc. etc.

Task 1: Get Setup

- Break into pairs
- Discuss who is "assigned" code?
- Need to get all sandboxes and IP addreses setup
- SSH key to zoned
- Subscribe to twingle-dev and twingle-commits

Task 2: Start Twingle

- Get debug-able approach in place
  - You want to avoid relaunching Twingle
  - Not possible for RDF changes
  - Some approaches hide menus on OS X

Task 2 con't

- Launch Mozilla browser from shell prompt
- Tools->Web Dev-> JavaScript Console
- Tools->Web Dev->DOM Inspector
- Enter chrome://twingle/content
- Look in Moz profile dir for twingle subdir
- Watch debug messages on console

Task 3: Change Button

- Open content/toolbar/toolbaroverlay.xul
- Find the first <button> element
- Change the value of its label attribute to your name
- Refresh (but not restart) your Twingle

Task 4: Unhide Column

- The navigation tree has two columns hidden
- Open navigation/navigationoverlay.xul
- Find second <treecolumn>
- Change hidden to “false”
- Refresh Twingle

Task 5: Rename Site

- Change the label on OSCOM site
- Open $profile/twingle/sites.rdf
- Edit <dc:title> for oscom.org
- Close all Mozilla windows and restart
- Note: we didn't change boilerplate

Task 6: Tab Label

- Goal: change the label on the Preview tab
- Open navigation/navigation.js
- Find first createElementNS (in openSelectedResource)
- This element is defined by Twingle in skin/twingle.css

Task 6 con't

- Open skin/twingle.css and find the selector for articleviewer
- This points to an XBL file at workspace/editorbindings.xml
- Purpose: an extensible type system of widgets
- The selector indicates we want the "articleviewerbinding" binding

Task 6 con't

- Open workspace/editorbindings.xml
- Find the xul:tab label="Preview"
- Change this and refresh Twingle
- Double-click on a resource in the tree

Blue Sky

- Improving identity/auth, FOAF
- Working offline
- Improve DAV support in OSCOM (past Level 2 and into DASL, Delta-V, ACL)
- Other views in navigation: Recent, People, etc.

Open Issues

- Eliminate toolbar?
- “Article” or “Document”? “Folder” or ??
- Display items in navigation, or just collections?