enfoldsystems

Security and Workflow

Andy McKay

Why this talk?

- This is the most asked for talk at the last Plone Conference
- ...*that nobody did*. Probably because it's not the most exciting topic.
- So here we go...

- The slides will be online at http://www.enfoldsystems.com (soon)

Contents

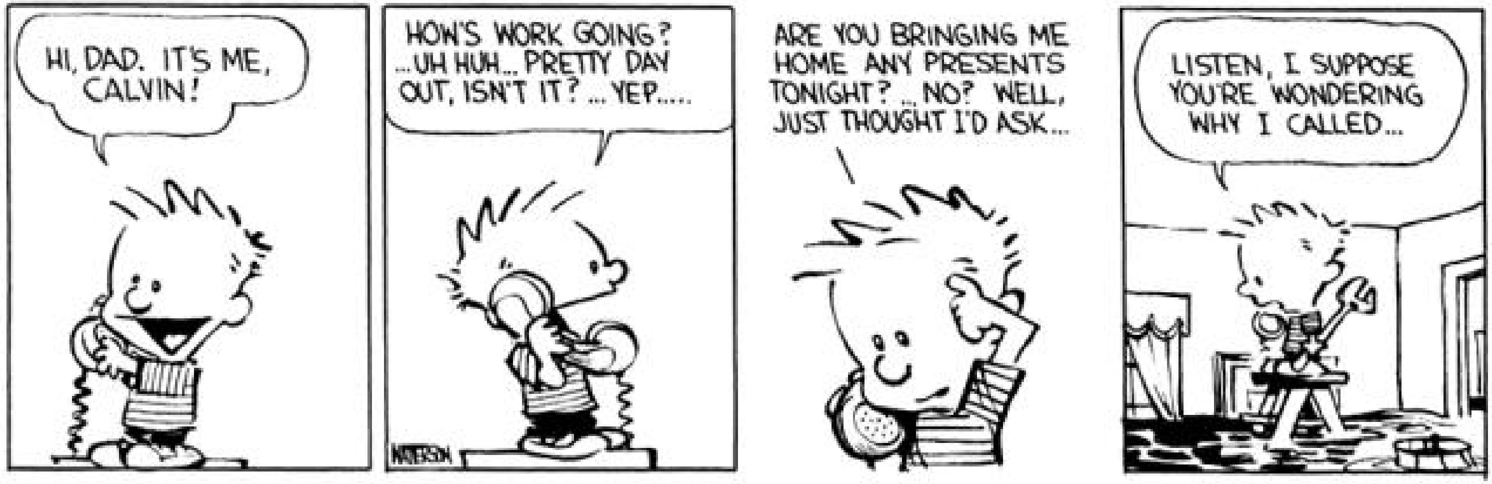
- Why Security *and* Workflow
- Zope Security
- Plone Workflow
- Security in Workflow
- Do's and Don'ts
  - – This is aimed more at beginners unfamilar with Plone security

Security and Workflow

- One of the key features of Zope:
  - Security
- One of the differentiating factors of Plone:
  - Workflow
- The two are quite heavily intertwined in Plone.

Before you start...

- Please
  - Do a backup
  - Remember the undo button...
- Chances are you will break your site


HI, DAD. IT'S ME, CALVIN!
HOW'S WORK GOING? ...UH HUH... PRETTY DAY OUT, ISN'T IT? ... YEP.....
WATTERSON
ARE YOU BRINGING ME HOME ANY PRESENTS TONIGHT? ... NO? WELL, JUST THOUGHT I'D ASK...
LISTEN, I SUPPOSE YOU'RE WONDERING WHY I CALLED...

Security

- Zope provides a complete security layer that Plone (as a Zope app.) uses
- On each and every call, Zope is going to:
  - Check who the user is
  - See if the user has the **right** to
    - Access that page
    - Call the methods on that page
    - Access content that the page accesses

Security (2)

- Main definitions
  - Users
    - A user is a particular user logged into a site
  - Roles
    - A role is a particular
    - Users are granted roles
  - Permissions
    - A permission to do something
    - Roles are given permissions

Users

- Users are defined in Zope
  - Live in a “User Folder” called “acl_users” and lives inside of ZODB
  - A Zope user is stored in the Zope instance, it is not related to the user that exists on the server
  - A User has:
    - A user name
    - A password
    - Some roles

Before people ask

- Yes a user folder can hook into other systems if appropriate code is written eg:
  - LDAPUserFolder pulls users out of an LDAP
  - ExUserFolder maintains plugins for a few things:
    - Smb, Postgres, Radius
  - Zope 3 specifically features Pluggable Authentication Service (PAS) specifically to allow different plugins

Tip

- Never, ever, ever change the root acl_users folder
- *Always* change the one in the sub folder or Plone site
  - It's not a question of if the user folder breaks, but when

Roles

- Roles are assigned to users there are the following roles by default:
  - Anonymous (Zope)
  - Authenticated (Zope)
  - Member (Plone/CMF specific)
  - Reviewer (Plone/CMF specific)
  - Manager (Zope)
  - Owner (Zope)
- The Member role is the default Plone role

What do roles mean?

- That depends upon the security settings but generally:
  - *Manager* = God
  - *Member* = Can add and edit content in certain folders. Can't publish it.
  - *Reviewer* = Can review other Members content. Can publish it.

Owner role

- This is a special role that is assigned to the person who created that object
  - Normally this person has more rights than someone else of the similar level
  - Eg: Bob and Bill are both members
  - But because Bob created a document, he is the owner and has more rights than Bill
  - Owner is assigned by Zope when the user creates something

Permissions

- Are assigned to roles
  - Let's take a look at a ZMI
  - Go to ZMI and click on *Security*
  - You'll see:
    - On horizontal: Roles in your site
    - On vertical: Permissions in your site
  - Where the intersection is checked is where the user has right to view

What do the permissions mean?

- That is a challenge
  - There is *no* documentation
  - The only real way is to go and read the code and see what is defined where to see what the options are
  - There are a few key ones:
    - Access Contents Information and View
    - Delete portal content
    - Modify portal content
    - Manage portal

Acquistion of Permissions

- Left hand column
  - “Acquire Permissions Settings”
  - Turns on or off acquisition of permission settings
  - If this is turned on
    - When its checking if Anonymous can View...
    - If in that object View is not selected, but Acquire... is then...
    - It will keep looking in each containing folder until it finds the permission or Acquire is off

Permission Acquisition

- This means you can go to the root of your Zope and Plone and..
  - Set the permissions for the whole site
  - For example if you allow Anonymous to “Add portal member” this means they can join the Plone site from anywhere in Plone
- However
  - Workflow often turns acquistion off

Members

- A member is a user with more information about them than a normal user eg:
  - Email address
  - First name
  - Last name
  - Etc...
  - A member is a super set of a user

Groups

- Specific to Plone thanks to GRUF
  - Allows you to put users into logical groups
  - Such as “Marketing”
- A group can also have
  - Data about the group (such as email)
  - Roles

Testing security

- Use 2 browsers
  - Log into the ZMI with one
  - Log into Plone with the other
- ZMI uses HTTP Auth
  - HTTP auth has no concept of sign out
- Plone uses Cookies

Plone – Control Panel

- The control panel allows you to:
  - Add and edit members
  - Add and edit groups
  - Assign members to groups
- So now it's easy
  - To make a “Reviewer Group” with the “Reviewer Role”...

Plone – Sharing tab

- Allows you to assign different *local roles*
  - A local role is a role for a particular owner or group for that folder and *everything below*
  - For example: Bill wants Bob to edit Bill's content, so goes to sharing tab and gives them Owner role...
- *Note*: PLIP 16 will allow you to limit to just the current folder

Plone areas

- Plone creates two key areas:
  - Members/...
  - groups/...
- These are folder's created for those members and groups to collaborate in
  - Members or groups are made the owner of those groups
  - So any member of group X can edit content in folder groups/X

When things go wrong

- Some products on the following pages
- You did a backup right?
- Looking at the ZMI can prove rather laborious

Some products for helping

- Verbose Security
  - http://hathaway.freezope.org/Software/VerboseSecurity
  - Tries to give you a detailed error message.
  - To use you have to go to *cookie_authentication* > *Auto-login page ID* and set this to blank.
  - Now go and raise an error again.
  - You'll want to turn this back the way it was before you go live...

Other products (2)

- Plone Debug (Collective)
  - Just adds user information to the left hand column, so you can see the rights for a user...
- In the *error_log* object, Unauthorized is turned off by default
  - So its not logged, go to *error_log* and remove Unauthorized from the log object...
  - you'll want to put this back the way it was before you go live...

Unit testing security

- This is the ideal situation, however I'm not sure how often this happens
  - To login in as *user* in code

```
from AccessControl.SecurityManager \
import newSecurityManager

uf = self.app.acl_users
user = uf.getUserById(portal_owner).__of__(uf)
newSecurityManager(None,   user)
```

Workflow

- So what's the connection?
- Well 90% of workflow's job is really about security (there is also notifications, but that's not in the scope)
  - Who can view what content and when?
  - Who can edit what content and when?
- This is the job of workflow

Workflow Overview

- The default workflow is
  - * Visible
    - viewable by any, not announced, editable by owner
  - Pending
    - viewable by any, not announced, edited by reviewers
  - Published
    - viewable by any, announced, editable by managers only
  - Private
    - viewable and editable by owners and managers

Workflow Permissions

- Set in the ZMI (or in Python):
  - To get there in the ZMI its:
    - *portal_workflow* > *contents* > *plone_worklow* > *states* > *[state]* > *permissions*
  - You can see permissions for each state
  - To have a permission managed in workflow go to:
    - *portal_workflow* > *contents* > *plone_worklow* > *permissions*

Permissions workflow manages

- Permissions managed
  - Access contents information
  - Change portal events
  - Modify portal content
  - View
- And recently
  - Webdav lock, unlock and access (although these look wrong, CVS ci anyone?)

Changing the permissions

- So supposing we wanted to allow Owners to edit published content
  - Go to *portal_workflow* > *contents* > *plone_worklow* > *states* > *published* > *permissions*
  - Check the box corresponding to *Manage Portal content* for *Owners*

Gotcha

- If the permission is managed by Workflow, then
  - When you transition something permissions will be changed
  - Don't try altering the permissions of an object managed by the workflow
- When you've changed workflow permissons
  - Go hit *Update Security Setttings*.
  - Since the change happens on transitions, things will be out of date.

Make private the default

- By default content is visible, which annoys people
- A different way is to set *private* to default
  - This allows people to post things, change them until they are right, suitable for extranets and internet sites
  - Go to *portal_workflow* > *contents* > *plone_worklow* > *states* and select the state to “Set Initial State”

Transition

- A transition is when you move from one state to the next
  - There are then security restrictions for when this can happen, you don't want just anyone to publish content
  - To view the security on a transition go to: *portal_workflow* > *contents* > *plone_worklow* > *transitions* > *[transition]*

Guard

- The security for a transition is called a guard
  - Visible at the bottom of a page
  - There are there options for a guard you can set either:
    - Permission(s):
      - Eg: Manage portal content
    - Role(s)
      - Eg: Manager
    - Expression
      - Any valid TALES

Remove publish step

- So if you wanted to remove the publish step you could give Owners the right to publish their own material
- To do this go to:
  - *portal_workflow* > *contents* > *plone_worklow* > *transitions* > *publish*
  - Add *Owner* in to the roles box
  - Any user who is the Owner **or** has “Review Portal Content” (Manager, Reviewer) can publish

Different workflows

- By default there is only one workflow for content type
  - But if the difference between workflow specs. is minor (say one tranisition or two) then you could
  - Use Guards to limit what transitions appear when, giving the appearance of two workflows but only having one
  - Eg following expression:
    - python: “public” in state_object.getPhysicalPath()
    - True for any object inside a folder called public

Workflow scripts

- Often you might want to move an object or do something in the workflow that requires more permission than user has
  - A worklow script is executed in as the user
  - You need a proxy role, go to script and give it a proxy role that is higher than the current one
  - Examples:
    - Moving content to another folder
    - Sending an email

Conclusion

- So we covered:
  - Zope security
  - Users and Members
  - Plone specific stuff
  - Workflow
- Questions?
- andy@enfoldsystems.com