

# Using Custom Post Types to Control Access

Building a collaborative web site by  
segmenting a site into areas of  
responsibility and control.

# Custom Post Types - CPTs

- What
  - Features and Attributes of CPTs
  - How they can be used for access control
- Why
  - The problem we are trying to solve
  - Why CPTs meet our needs
- How
  - Implementing CPTs
  - Simple Case Study

# Background

- A bit about the organization
  - All about bicycle touring
  - Tours are developed and run by volunteers
  - Local groups in several various cities across Canada
- What we used to have
- Where we wanted to go
  - Modern web software
  - Break the single person bottle neck

# Breaking the Single Person Bottleneck

- Different approaches
  - Make everyone an admin
  - Multiple contributors – central publication
  - Multiple restricted editors
- Restrict access for security purposes
- Restrict access to enhance simplicity

# Simple Restricted Dashboard

The screenshot displays a WordPress dashboard with a dark sidebar on the left. The sidebar contains the following menu items: Dashboard, Comox Posts (highlighted), All Comox Posts, Add New, Profile, and Collapse menu. The main content area is titled 'Comox Posts' and includes an 'Add New' button and a 'Screen Options' dropdown. Below the title, there is a user notification: 'Switch back to Max McClanahan (max.mcclanahan)'. A search bar is present with the text 'Search Comox Posts'. The main content shows a list of posts with the following controls: 'All (1) | Published (1)', 'Bulk Actions' dropdown, 'Apply' button, 'All dates' dropdown, and 'Filter' button. The table below has three columns: 'Title', 'Date', and 'Type'. The first row contains a checkbox, the text 'First Post', the date 'Published 2016/08/23', and the text 'Comox Post'. The second row contains a checkbox, the text 'Title', the text 'Date', and the text 'Type'. At the bottom of the table, there are 'Bulk Actions' and 'Apply' buttons, and a '1 item' indicator on the right.

Dashboard

Comox Posts

All Comox Posts

Add New

Profile

Collapse menu

Comox Posts [Add New](#) Screen Options ▾

[Switch back to Max McClanahan \(max.mcclanahan\)](#)

Search Comox Posts

All (1) | Published (1)

Bulk Actions ▾ Apply All dates ▾ Filter 1 item

<input type="checkbox"/> Title	Date	Type
<input type="checkbox"/> <a href="#">First Post</a>	Published 2016/08/23	Comox Post
<input type="checkbox"/> Title	Date	Type

Bulk Actions ▾ Apply 1 item



# Why use CPTs

- Alternative architectures
  - Restricted categories
    - Addresses only security requirements not simplicity
    - Doesn't fully solve security issues
    - Much less elegant than capabilities
  - Multi-site installation
    - Would probably work
    - Adds unnecessary complexity

# Users, Roles and Capabilities

- Users
  - Must have one or more roles
- Roles
  - Standard System Roles
    - Administrator, Editor, Author, Contributor, Subscriber
  - Custom Roles
    - Club Document Editor, Rides Director
- Capabilities
  - read, delete\_posts, edit\_posts



# What are Post Types

- Great Concept – bad name
- Default System Post Types (i.e. non-custom post types)
  - Post
  - Page
  - Attachment
  - Revision
  - Navigation Menu Item

# Post Type Common Features

- A set of fields
- A set of capabilities
- A set of taxonomies
- A set of attributes
  - Hierarchical ?
  - Archive page ?

# What are Taxonomies

- A list of terms
- Can have a parent (category) or not (tags)
- Associated to 0 or more CPTs (many-to-many)
- Capabilities
  - Manage\_terms
  - Edit\_terms
  - Delete\_terms
  - Assign\_terms

# Case Study

- Local Editor
  - Comox\_editor, Victoria\_Editor, Ottawa\_editor
  - CPT for each local editor restricts his area of authority
- Club Document Editor
  - CPT to restrict area of authority
  - Custom Taxonomy to categorise different document types
- Ride Director
  - CPT to restrict area AND add custom fields
  - Uses built-in tax taxonomies

# Plugins Employed

- Pods – Content Types and Fields
  - “Custom Post Type UI” is a possible alternative
- Members
  - “Roles & Capabilities” component of Pods is an alternative
- Post Type Switcher
  - Essential for converting an existing site
- User Switching
  - Not strictly required, but very useful for testing

# Setting up a Local Editor

- Create the CPT “Comox Posts”
  - PODS plugin
- Create the “Comox Editor” role
  - Members plugin
  - Users | new Role
- Setup menus to Reference CPT’s
  - Create a *Custom Link* menu item
  - URL: *site-url/?post\_type=comox\_post*

# Create the CPT

Add New Pod [← Back to Manage](#)

## 1 Create or Extend

## 2 Configure

Pods are content types that you can customise and define fields for based on your needs. You can choose to create a Custom Post Type, Custom Taxonomy, or Custom Settings Pages for site-specific data. You can also extend existing content types like WP Objects such as Post Types, Taxonomies, Users, or Comments.

Not sure what content type you should use? Check out our [Content Type Comparison](#) to help you decide.




### Create New

Create entirely new content types using Post Types, Taxonomies, or Custom Settings Pages.

### Extend Existing

Extend any existing content type within WordPress, including Post Types (Posts, Pages, etc), Taxonomies (Categories, Tags, etc), Media, Users, or Comments.



# Create the CPT cont.

Add New Pod

[← Back to Manage](#)

1 Create or Extend ✓

2 Configure

Creating a new Content Type allows you to control exactly what that content type does, how it acts like, the fields it has, and the way you manage it.

## Create a New Content Type

Content Type ?

Custom Post Type (like Posts or Pages) ▼

Singular Label ?

Comox Post

Plural Label ?

Comox Posts

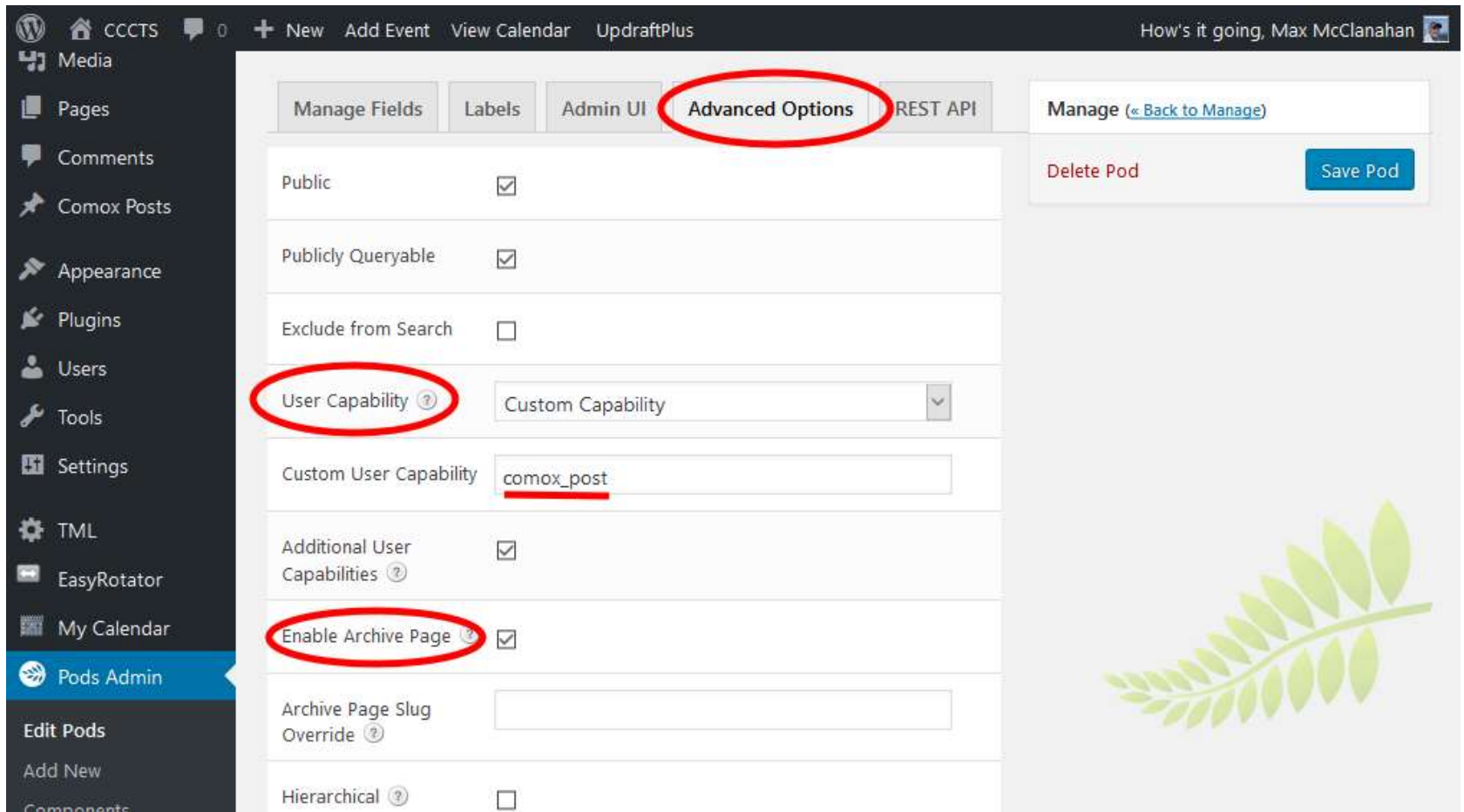
[Advanced +](#)

**Start Over**

**Next Step**



# Create the CPT cont.

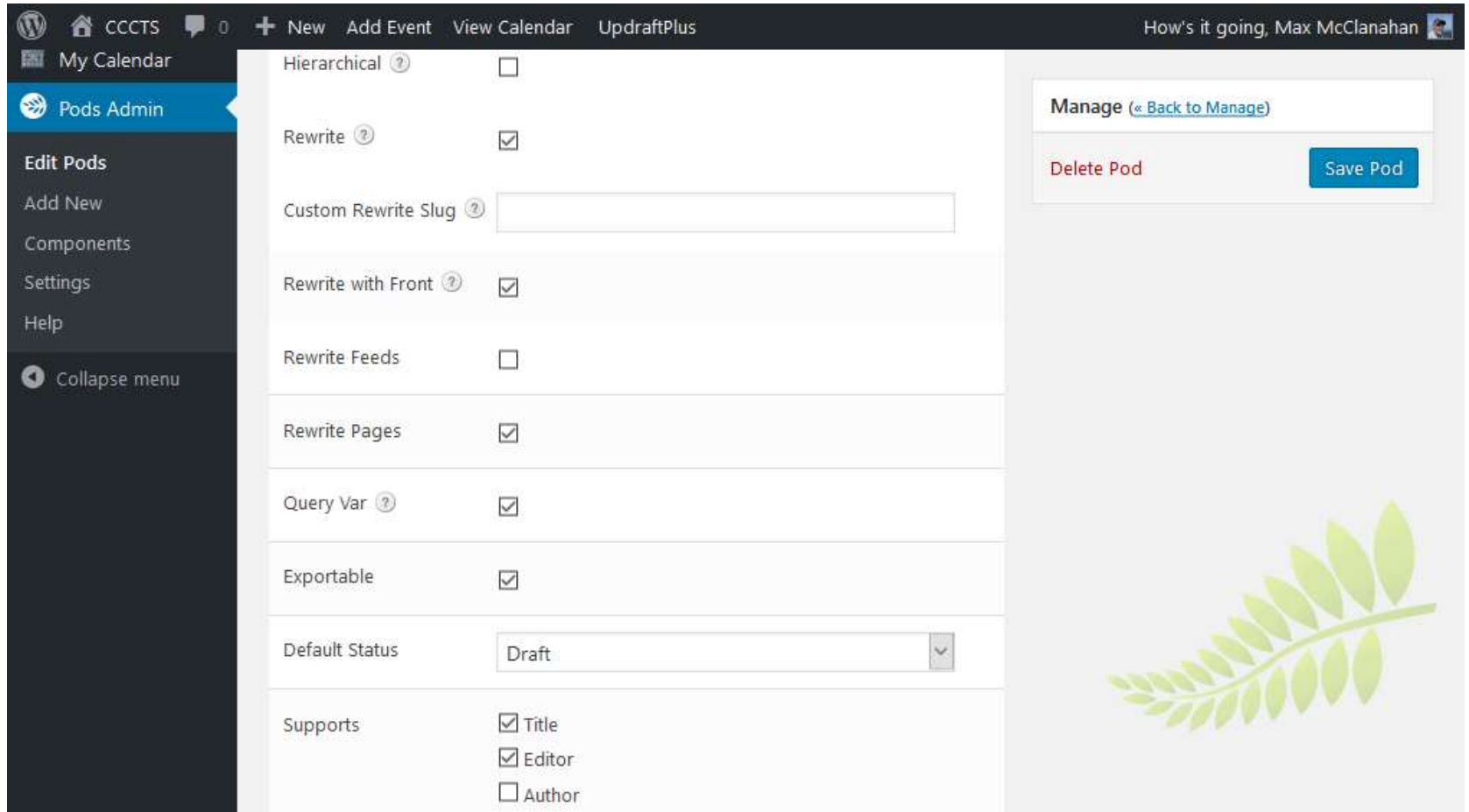


The screenshot displays the WordPress Pods Admin interface. The top navigation bar includes links for 'New', 'Add Event', 'View Calendar', and 'UpdraftPlus'. The user's name 'Max McClanahan' is visible in the top right. The left sidebar contains a menu with items like 'Media', 'Pages', 'Comments', 'Comox Posts', 'Appearance', 'Plugins', 'Users', 'Tools', 'Settings', 'TML', 'EasyRotator', 'My Calendar', 'Pods Admin', 'Edit Pods', and 'Add New Components'. The main content area shows the configuration for a Custom Post Type (CPT). The 'Advanced Options' tab is selected and circled in red. The configuration includes the following options:

- Public**:
- Publicly Queryable**:
- Exclude from Search**:
- User Capability**:  Custom Capability (circled in red)
- Custom User Capability**: comox\_post
- Additional User Capabilities**:
- Enable Archive Page**:  (circled in red)
- Archive Page Slug Override**:
- Hierarchical**:

On the right side, there are buttons for 'Delete Pod' and 'Save Pod', and a 'Manage' button with a link to 'Back to Manage'. A green leaf graphic is visible in the bottom right corner of the interface.

# Create the CPT cont.



The screenshot displays the WordPress Pods Admin interface for configuring a Custom Post Type (CPT). The top navigation bar includes the WordPress logo, site name 'CCCTS', and navigation links: '+ New', 'Add Event', 'View Calendar', and 'UpdraftPlus'. The user's name 'Max McClanahan' is visible in the top right. The left sidebar contains the 'Pods Admin' menu with options: 'Edit Pods', 'Add New', 'Components', 'Settings', 'Help', and 'Collapse menu'. The main content area shows the following configuration options:

- Hierarchical
- Rewrite
- Custom Rewrite Slug
- Rewrite with Front
- Rewrite Feeds
- Rewrite Pages
- Query Var
- Exportable
- Default Status: Draft (dropdown menu)
- Supports:  Title,  Editor,  Author

On the right side, there is a 'Manage' section with a link to 'Back to Manage', a 'Delete Pod' button, and a 'Save Pod' button. A large green leaf graphic is positioned in the bottom right corner of the interface.

# Create the CPT cont.

WordPress dashboard header: CCCTS, 0 notifications, + New, Add Event, View Calendar, UpdraftPlus. User: How's it going, Max McClanahan.

Feature list:

- Editor
- Author
- Featured Image
- Excerpt
- Trackbacks
- Custom Fields
- Comments
- Revisions
- Page Attributes
- Post Formats


Advanced Supports ?

Built-in Taxonomies:

- Categories (category) (category)
- Link Categories (link\_category) (link\_category)
- Categories (mc-event-category) (mc-event-category)
- Tags (post\_tag) (post\_tag)

Manage [« Back to Manage](#)

Delete Pod Save Pod



Thank you for creating with [WordPress](#). Version 4.6

# Create “Comox Editor” Role

The screenshot shows the WordPress user role editor interface. The sidebar menu on the left includes: Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users (highlighted), All Users, Add New, Your Profile, Roles, Add New Role, Tools, and Settings. The main content area is titled "Edit Capabilities: Comox Posts". A red arrow points to the "Grant" column header. The table lists various capabilities with checkboxes for Grant and Deny. The "Comox Posts" category in the sidebar is circled in red.

	Capability	Grant	Deny
General	edit_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Posts	edit_others_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pages	publish_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Media	read_private_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pods	delete_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pod Fields	delete_private_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	delete_published_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Comox Posts	delete_others_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	edit_private_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	edit_published_comox_posts	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Appearance	Capability	Grant	Deny

On the right side, there is a "Denied: 0" section with an "Add Role" button, and a "Custom Capability" section with an "Add New" button.

# More on Users and Role

- Add the new CPT capabilities to the Administrator
- Create new user “Joe Comox”
  - Optional user for testing
  - Assign “Comox Editor” role
- Confirm Role and CPT are working
  - Switch user to “Joe Comox”
  - Create a new *Comox Post*

# Creating a *Comox Post*

The screenshot displays the WordPress Comox Post editor interface. At the top, the navigation bar includes the WordPress logo, site name 'CCCTS', a '+ New' button, and a 'View Calendar' link. The user's name 'Joe Comox' is visible in the top right corner. The left sidebar contains navigation options: 'Dashboard', 'Comox Posts' (highlighted), 'All Comox Posts', 'Add New', 'Profile', and 'Collapse menu'. The main content area is titled 'Add New Comox Post' and shows the user 'Max McClanahan' with a 'Switch back to Max McClanahan' link. The post title is 'First Post', and its permalink is 'http://localhost/cccts/comox-post/first-post/'. The editor is in 'Visual' mode, showing a rich text editor with a menu bar (File, Edit, Insert, View, Format, Table, Tools) and a toolbar with icons for bold, italic, underline, quote, list, link, unlink, table, and image. The text in the editor reads 'This is my first Comox Post'. On the right, the 'Publish' sidebar contains buttons for 'Save Draft', 'Preview', and 'Publish', along with status and visibility information: 'Status: Draft', 'Visibility: Public', and 'Publish immediately'.

# Menu Link to *Comox Post* Archive

The screenshot displays the WordPress Admin Dashboard's 'Edit Menu' interface. The top navigation bar includes links for 'Media', 'Pages', 'Comments', 'Comox Posts', 'Appearance', 'Themes', 'Customize', 'Widgets', 'Menu', 'Header', 'Editor', 'Plugins', 'Users', 'Tools', 'Settings', and 'TML'. The 'Appearance' menu is expanded, and a red arrow points to the 'Menu' option. The 'Edit Menu' interface shows a 'Menu Name' field with the value 'main' and a 'Save Menu' button. The 'Menu Structure' section contains a list of menu items with their respective configurations:

Menu Item	Type
Home	Page
Help <i>sub item</i>	Category
Club Information	Page
About the Club <i>sub item</i>	Page
Membership <i>sub item</i>	Page

The 'Custom Links' section is highlighted with a red arrow. It contains the following fields:

- URL: `?post_type=comox_post`
- Link Text: Comox Valley Group
- Buttons: Add to Menu

# What's Next?

- Use Pods “duplicate” functionality to create additional similar CPT's
- Create a corresponding role for each CPT
- Add users to roles
- Add menu items for each CPT archive
- Deploy



# Club Document Editor Case Study

- Slightly more complicated example than the Local Editor
- Adds a custom taxonomy to categorize different types of documents
- First we will Create the custom taxonomy using Pods
- Then we will create our “Club Document” CPT as in the previous example

# Create a Custom Taxonomy

WordPress dashboard header: CCCTS, 0 comments, + New, Add Event, View Calendar, UpdraftPlus, How's it going, Max McClanahan


Dashboard sidebar: Dashboard, Posts, Media, Pages, Comments, Comox Posts, Ottawa Posts, Victoria Posts, Appearance, Plugins, Users, Tools, Settings, TML

## Add New Pod [← Back to Manage](#)

**1** Create or Extend ✓ **2** Configure

Creating a new Content Type allows you to control exactly what that content type does, how it acts like, the fields it has, and the way you manage it.

### Create a New Content Type

Content Type ?	Custom Taxonomy (like Categories or Tags) ▼	
Singular Label ?	Document Type	
Plural Label ?	Document Types	

[Advanced +](#)

**Start Over** **Next Step**

# Create Taxonomy cont.

The screenshot shows the WordPress Admin Dashboard with the 'Create Taxonomy' settings page. The top navigation bar includes 'CCCTS', 'New', 'Add Event', 'View Calendar', and 'UpdraftPlus'. The user's name 'Max McClanahan' is visible in the top right. The left sidebar shows the 'Settings' menu. The main content area displays the 'Create Taxonomy' settings for a taxonomy named 'document type'. A success message at the top reads 'Success! Pod created successfully.' The settings are organized into tabs: 'Manage Fields', 'Labels', 'Admin UI', 'Advanced Options', and 'REST API'. The 'Advanced Options' tab is active, showing the following settings:

- Public:
- Hierarchical:  (highlighted with a red arrow)
- Label: Parent Document Type:
- Label: Parent:
- Label: No Document Types:
- Rewrite:
- Custom Rewrite Slug:

On the right side, there is a 'Manage' section with a 'Back to Manage' link, a 'Delete Pod' button, and a 'Save Pod' button. A green leaf graphic is visible in the bottom right corner of the settings area.

# Create Taxonomy cont.

The screenshot displays the WordPress Pods Admin interface. The top navigation bar includes the WordPress logo, a home icon, the site name 'CCCTS', a notification icon with '0', and links for '+ New', 'Add Event', 'View Calendar', and 'UpdraftPlus'. The user's name 'How's it going, Max McClanahan' is visible in the top right.

The left sidebar contains the following menu items: 'TML', 'EasyRotator', 'My Calendar', 'Pods Admin' (highlighted in blue), 'Edit Pods', 'Add New', 'Components', 'Settings', 'Help', and 'Collapse menu'.

The main content area shows the configuration for a taxonomy. The 'User Capability' dropdown is set to 'Custom Capability', and the 'Custom User Capability' field contains 'document\_type'. Two red arrows point to these fields. Other options include 'Allow Front Prepend' (checked), 'Hierarchical Permalinks' (checked), 'Query Var' (unchecked), 'Remember order saved on Post Types' (unchecked), and a text input for 'Function to call when updating counts'.

The 'Associated Post Types' section is underlined in red and lists several options with checkboxes: 'Comox Posts (comox\_post)', 'Events (mc-events)', 'Ottawa Posts (ottawa\_post)', 'Pages (page)', 'Posts (post)', and 'Rides (ride)'. None are currently selected.

On the right side, there is a 'Manage (< Back to Manage)' link, a 'Delete Pod' link, and a 'Save Pod' button.

A green leaf graphic is visible in the bottom right corner of the interface.

# Create “Club Document” CPT

WordPress admin interface showing the "Edit Pod: club\_document" settings page. The page includes a success message: "Success! Pod created successfully." The settings are organized into tabs: Manage Fields, Labels, Admin UI, Advanced Options, and REST API. The "Advanced Options" tab is active, showing the following settings:

- Public:
- Publicly Queryable:
- Exclude from Search:
- User Capability: Custom Capability (dropdown menu) ← Red arrow
- Custom User Capability: club\_document
- Additional User Capabilities:
- Enable Archive Page:  ← Red arrow

On the right side, there is a "Manage" button with a link to "Back to Manage", a "Delete Pod" button, and a "Save Pod" button. A green leaf graphic is visible in the bottom right corner of the settings area.

# Create “Club Document” CPT cont.


WordPress dashboard header: CCCTS, 0 notifications, + New, Add Event, View Calendar, UpdraftPlus. User: How's it going, Max McClanahan.

**Supports**

- Title
- Editor
- Author
- Featured Image
- Excerpt
- Trackbacks
- Custom Fields
- Comments
- Revisions
- Page Attributes
- Post Formats


**Advanced Supports** [?](#)

**Built-in Taxonomies**

- Categories (category) (category)
- Document Types (document\_type) (document\_type) 
- Link Categories (link\_category) (link\_category)
- Categories (mc-event-category) (mc-event-category)
- Tags (post\_tag) (post\_tag)

**Manage** ([← Back to Manage](#))

[Delete Pod](#) [Save Pod](#)



# Create Role and User

- Add new role: Club Document Editor
  - Assign all *Club Document* capabilities
- Assign all *Club Document* capabilities to administrator
- Create new user: Sally Secretary
  - Assign *Club Document Editor* Role
- Populate our *Document Types* Taxonomy
  - Add: “Board Minutes”, “General Meeting Minutes” and “uncategorized”

# Test: Add New *Club Document*

The screenshot displays the WordPress interface for adding a new club document. The top navigation bar includes the WordPress logo, a home icon, the site name 'CCCTS', a '+ New' button, and a 'View Calendar' link. The user's name 'Sally Secretary' is visible in the top right corner. The left sidebar contains navigation options: 'Dashboard', 'Club Documents' (highlighted), 'All Club Documents', 'Add New', 'Profile', and 'Collapse menu'. The main content area is titled 'Add New Club Document' and features a text input field for the title with the placeholder 'Enter title here'. Below the title field is a rich text editor with tabs for 'Visual' and 'Text (HTML)'. The editor's toolbar includes options for bold, italic, underline, quote, list, link, unlink, table, and other formatting tools. On the right side, the 'Publish' sidebar contains buttons for 'Save Draft' and 'Preview', along with status and visibility information: 'Status: Draft', 'Visibility: Public', and 'Publish immediately'. Below this is a 'Document Types' section with two tabs: 'All Document Types' and 'Most Used'. Under the 'All Document Types' tab, there are three checkboxes: 'Board Minutes', 'General Meeting Minutes', and 'Uncategorized', all of which are circled in red.



# Why Can't Sally Select a Document Type?

- Hint: If I switch back to my administrator role everything works fine.
- But we assigned Sally all *Club Document* capabilities!
- *Document Types* is a taxonomy not a CPT
- Where do we set taxonomy capabilities?

# Setting Taxonomy Capabilities

The screenshot shows the WordPress Admin interface for editing a role. The role is named "Club Document Editor" and has the slug "club\_document\_editor". The "Edit Capabilities" section is titled "Taxonomies" and contains a table of capabilities. The following table represents the data in this section:

	Capability	Grant	Deny
General	manage_categories	<input type="checkbox"/>	<input type="checkbox"/>
Posts	assign_categories	<input type="checkbox"/>	<input type="checkbox"/>
Pages	assign_post_tags	<input type="checkbox"/>	<input type="checkbox"/>
Media	manage_document_type_terms	<input type="checkbox"/>	<input type="checkbox"/>
Pods	edit_document_type_terms	<input type="checkbox"/>	<input type="checkbox"/>
Pod Fields	delete_document_type_terms	<input type="checkbox"/>	<input type="checkbox"/>
Club Documents	assign_document_type_terms	<input checked="" type="checkbox"/>	<input type="checkbox"/>

On the right side of the interface, the "Role" summary shows 1 user, 12 granted capabilities, and 0 denied capabilities. There are "Delete" and "Update" buttons. Below this is a "Custom Capability" section with an empty input field and an "Add New" button.

# Taxonomy Capabilities

- Grant *assign\_document\_type\_terms* to *Club Document Editor* role
- Grant all 4 *document\_type* taxonomy capabilities to administrator
- Flush the Pods cache:
  - *Pods Admin | Settings | Clear Pods Cache*

# Menu Links to Archives

The screenshot shows the WordPress dashboard with the 'Appearance' menu selected. The 'Document Types' section is expanded, showing a list of document types with checkboxes. 'Board Minutes' and 'General Meeting Minutes' are checked. The 'Menu Structure' panel is also visible, showing a list of menu items with their respective configurations.

**Document Types**

- Board Minutes
- General Meeting Minutes

[Select All](#) [Add to Menu](#)

**Menu Structure**

Drag each item into the order you prefer. Click the arrow on the right of the item to reveal additional configuration options.

- Home Page
- Help *sub item* Category
- Club Information Page
- About the Club *sub item* Page
- Membership *sub item* Page
- Club Contacts *sub item* Page
- Board Minutes *sub item* Document Type
- General Meeting Minutes *sub item* Document Type
- Local Groups Page

# Ride Director Case Study

- Our most complicated CPT
- Numerous custom field using table based storage
  - But that is a topic for another day
- Custom post type that uses the built-in taxonomies *category* and *tags*

# Create Rides CPT

The screenshot shows the WordPress admin dashboard with the 'Pods Admin' menu item selected. The main content area displays the configuration for a new Custom Post Type (CPT) named 'Rides'. The 'Default Status' is set to 'Draft'. The 'Supports' section is expanded, showing a list of features that can be enabled for the CPT. The 'Built-in Taxonomies' section is also expanded, showing a list of taxonomies that can be associated with the CPT. Two red arrows point to the 'Categories (category) (category)' and 'Tags (post\_tag) (post\_tag)' options, indicating they are selected.

WordPress Admin Dashboard: CCCTS, 0 messages, + New, Add Event, View Calendar, UpdraftPlus. User: How's it going, Max McClanahan.

Left sidebar menu: kides, Appearance, Plugins, Users, Tools, Settings, TML, EasyRotator, My Calendar, **Pods Admin**, Edit Pods, Add New, Components, Settings, Help, Collapse menu.

Main Content Area:

- Default Status: Draft
- Supports:
  - Title
  - Editor
  - Author
  - Featured Image
  - Excerpt
  - Trackbacks
  - Custom Fields
  - Comments
  - Revisions
  - Page Attributes
  - Post Formats
- Advanced Supports:
- Built-in Taxonomies:
  - Categories (category) (category) ←
  - Document Types (document\_type) (document\_type)
  - Link Categories (link\_category) (link\_category)
  - Categories (mc-event-category) (mc-event-category)
  - Tags (post\_tag) (post\_tag) ←

Right sidebar: Manage (← Back to Manage), Delete Pod, Save Pod.

# Rides: Roles & Capabilities

- Create new “Rides Director” role
  - Grant all *Rides* capabilities
- Grant all *Rides* capabilities to Administrator
- Create user “Ron Rider” with *Ride Director* role.

# Test: Add New Ride

The screenshot shows the WordPress dashboard for a user named Ron Rider. The main content area is the 'Add New Ride' editor, which is currently in 'Draft' status and set to 'Public' visibility. The editor includes a rich text toolbar with options for bold, italic, underline, quote, list, link, and unlink. The 'Categories' section is expanded, showing a list of categories: 'Hub and Spokes', 'Proposed', 'Tours', and 'Uncategorised'. The 'Hub and Spokes' category is circled in red. Below the categories is a 'Tags' section, which is also circled in red. The 'Publish' button is visible in the top right corner of the editor area. The word count at the bottom left is 0.

Word count: 0



# We Have Been Here Before

- Why can't Ron assign *categories* or *tags*?
- Again we have a capability problem
- Same problem (slightly) different solution
  - *Create custom capabilities*
    - *assign\_categories, assign\_post\_tags*
  - *Grant both capabilities to Rides Director & Administrator*
  - *Bind the taxonomy using a filter in functions.php*
  - *Flush the Pods cache*

# Creating Custom Capabilities

WordPress Admin Dashboard: CCCTS | 0 New | Add Event | View Calendar | UpdraftPlus | How's it going, Max McClanahan

## Edit Role [Add New](#)

Role: ride\_director

### Edit Capabilities: Taxonomies

+ All	Capability	Grant	Deny
General	manage_categories	<input type="checkbox"/>	<input type="checkbox"/>
Posts	assign_categories	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Posts	assign_post_tags	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pages	assign_post_tags	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Media	manage_document_type_terms	<input type="checkbox"/>	<input type="checkbox"/>
Pods	edit_document_type_terms	<input type="checkbox"/>	<input type="checkbox"/>
Pod Fields	delete_document_type_terms	<input type="checkbox"/>	<input type="checkbox"/>
Pod Fields	assign_document_type_terms	<input type="checkbox"/>	<input type="checkbox"/>
Club Documents	Capability	Grant	Deny

**Role Summary:**  
Users: 1  
Granted: 13  
Denied: 0  
[Delete](#) [Update](#)

**Custom Capability:**  
assign\_post\_tags  
[Add New](#)

# Binding Built-in Taxonomy Capabilities

```
/* In order to use a builtin taxonomy inside a CPT we need
 * to setup capabilities for the taxonomy that are enabled
 * in the role for that CPT
 */
```

```
add_action('init', 'set_builtin_tax_caps');
function set_builtin_tax_caps() {
    $tax = get_taxonomy('post_tag');
    $tax->cap->assign_terms = 'assign_post_tags';
    $tax = get_taxonomy('category');
    $tax->cap->assign_terms =
        'assign_categories';
}
```

# Summary

- Post Types (built-in or custom) each have an associated set of capabilities.
- By taking advantage of that functionality we can segment our site into discreet areas of responsibility
- Restrict access to only essential functionality
- Simplifies the interface by limiting to a specific area

Hopefully this discussion has provided a better understanding of how Custom Post Type can provide the infrastructure for building a collaborative web site.

**QUESTIONS?**