



Extending WooCommerce Like A Pro



wooooo!!!

WHO IS CHRIS?

- Back End Developer at Modern Tribe (we're hiring) <https://tri.be>
- Husband & Father
- Craft Beer Lover
- Musician
- Beach Bum



WHO IS CHRIS?

- Works on enterprise level WooCommerce sites every day
- Five years experience focused on WooCommerce
- ... and still haven't mastered it



@CHRISFLANNY
WHOISCHRIS.COM

FIRST TIME GIVING THIS TALK
TRYING NEW THINGS

THIS TALK IS PHP FOCUSED

INTERRUPT ME

It's impossible to present on just WooCommerce development

A lot of this can be applied to WordPress development in general

- Approaching WooCommerce - what's the goal?
- Building A Framework - make a mini app
- Finding What You're Looking For - hookin' up
- Let's Code - demo
- Staying Compatible - avoiding disaster

WHAT'S THE GOAL



HAVE CLEAR END PRODUCT

- Wireframe and strategy before writing any code
- Make a list of where and how your code will touch WooCommerce

WHAT DO WE TOUCH

- Product/Variation Meta
- Order Meta
- Order Item Meta
- Customer Meta
- Templates
- Currency
- Pricing/Taxes
- Fees/Coupons
- Reports

WHAT DO WE CREATE

- Gateways
- Custom Order Status(es)
- Custom Tables
- WP REST API Endpoints
- Imports/Exports
- Changes In Checkout Flow
- Problems

TOUCHING STUFF

- Touching one or two Woo things? Build a simple plugin

SIMPLE PLUGIN

- PHP 5.2 compatible
- Use a single plugin file if snippets
- No reason to make a child theme for small UI changes when hooks are available

SIMPLE PLUGIN

Let's see an example

BUILDING A FRAMEWORK



Keep S**T Organized!

THE TRIBE WAY

- I do it the Tribe way. You do it your way as long as it makes sense.
- Square One
- Containers /
Dependency Injections



KEEP THE THEME IN THEMES

- If you are theming your shop then do it in a theme or child-theme
- Use the WooCommerce template structure
- Keep logic out of template files and in a plugin

NAMESPACE AND AUTOLOAD

- If you can...
- If you want to extend ANYTHING like a pro then you need to do it the right way
- Highly important if you work with a team and just as important if you work solo

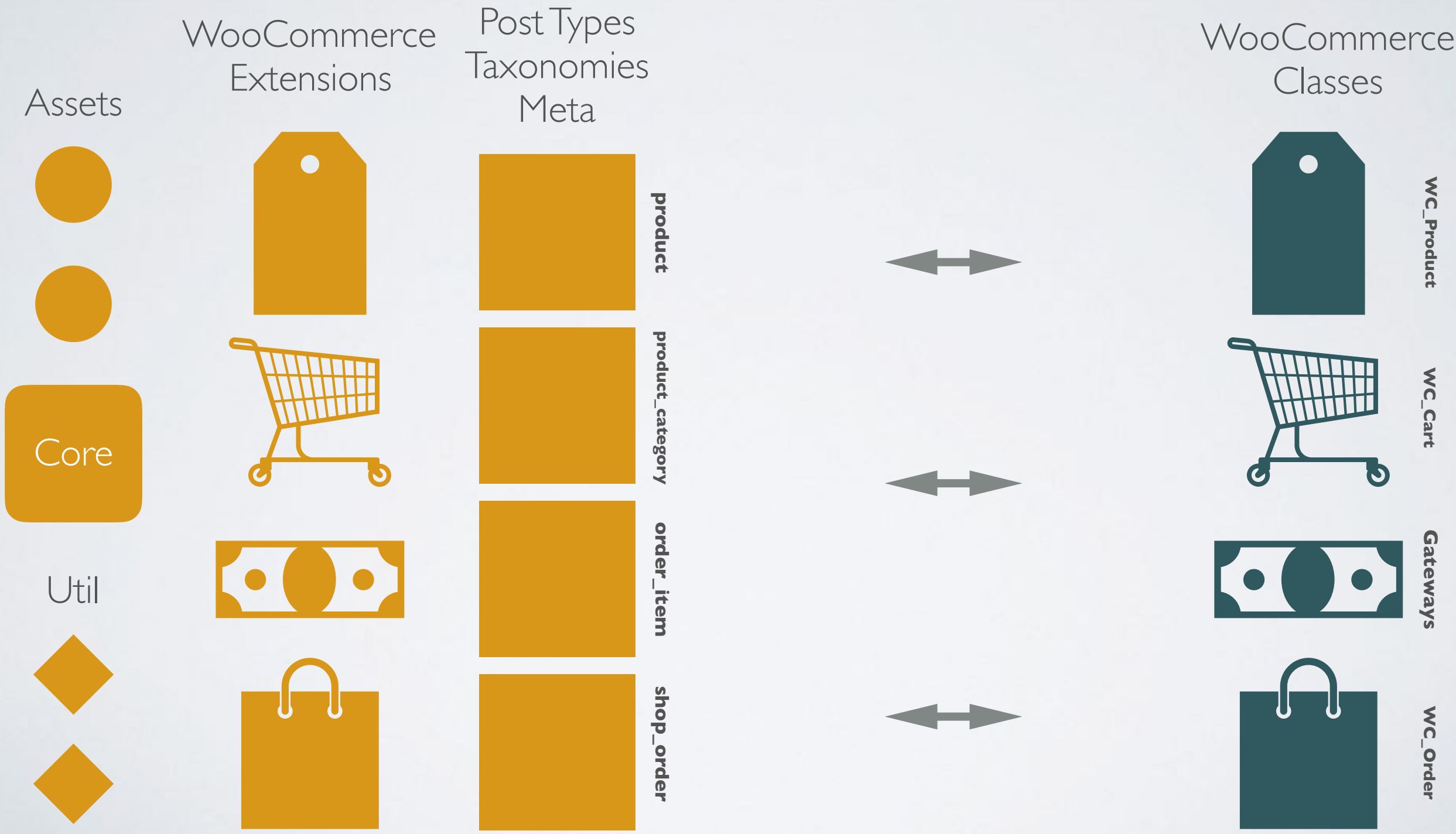
USE VERSION CONTROL

- Even in solo development using version control gives huge advantages
 - Historical data
 - Not tied to a single machine
 - Deploy code, never FTP
 - Great skill to have
- Commit the entire WP install if a single project/site, install WP with composer

DOCUMENT EVERYTHING

- Take the time to mark up all of your code.
- Small methods, simple parameter and return definitions are fine
- Explain complex logic
- Explain odd classes
- Document functions with the hooks that are calling them

STRUCTURE



STRUCTURE

Let's see an example

HOOBIN' UP



SO MANY OPTIONS

- Nail down exactly what you want to happen when you want it to happen
- Find the most appropriate filter or action
- Think thoroughly what's happening before and after your code during execution

PRIORITY

- Any number of functions could be running your data through it
- Getting odd results run a search on the codebase for the filter
- Dynamic hooks make this a lot more difficult
class-wc-webhook.php
`add_action( $hook, array( $this, 'process' ) );`
- Good reason to keep 3rd party Woo plugins limited

MULTIPLE PATHS

- Woo typically calls the same hooks via ajax or submitted form data but not always
- Biggest multi-path culprit I've found is the [shortcode] checkout form

```
// Runs during order_review_update WooCommerce ajax when checking out.  
add_action( 'woocommerce_after_checkout_validation', [ $this, 'validate' ] );  
// Runs after form submission from shortcode form.  
add_action( 'woocommerce_before_pay_action', [ $this, 'validate_shortcode' ] );
```

```
$products = WC()->cart->get_cart();  
if ( isset( $_REQUEST['pay_for_order'], $_REQUEST['key'] ) ) {  
    $order = new \WC_Order( wc_get_order_id_by_order_key( $_REQUEST['key'] ) );  
    $products = $order->get_items();  
}
```

MULTIPLE PATHS

- Woo doesn't filter out products with hidden catalog visibility outside of main shop loop
- Important to remember when using the REST API for front end display

```
add_action( 'pre_get_posts', [ $this, 'hidden_search_query_fix' ] );
...
public function hidden_search_query_fix( $query = false ) {
    if ( ! is_admin() && isset( $query->query['post_type'] ) && $query->query['post_type'] === 'product' ) {
        $tax_query = $query->get( 'tax_query' );
        $tax_query[] = [
            'relation' => 'OR',
            [
                'taxonomy' => 'product_visibility',
                'field'     => 'name',
                'terms'     => 'exclude-from-catalog',
                'operator' => 'NOT IN',
            ],
            [
                'taxonomy' => 'product_visibility',
                'field'     => 'name',
                'terms'     => 'exclude-from-catalog',
                'operator' => '!=',
            ],
        ],
    ];

    $query->set( 'tax_query', $tax_query );
    }
}
```

REPORTS

- Possibly my biggest issue with Woo core
- Giant SQL queries requiring complex injections to filter results

CLOSURES FOR FILTER

```
$filter_english_state_name = function( $states ) use ( $address ) {  
    $states[ $address['country'] ][ $address['state'] ] = $address['state'];  
    return $states;  
};  
  
add_filter( 'woocommerce_states', $filter_english_state_name );  
  
unload_textdomain( 'woocommerce' );  
  
WC()->countries->states = null;  
  
WC()->countries->load_country_states();  
  
$formatted_address = WC()->countries->get_formatted_address( $address );  
  
remove_filter( 'woocommerce_states', $filter_english_state_name );  
  
WC()->load_plugin_textdomain();
```


SO THAT WAS COOL

...but entirely unnecessary

```
unload_textdomain( 'woocommerce' );
```

```
WC()->countries->states = null;
```

```
WC()->countries->load_country_states();
```

```
$formatted_address = WC()->countries->get_formatted_address( $address );
```

```
WC()->load_plugin_textdomain();
```

```
WC()->countries->load_country_states();
```

USE THE RIGHT TOOLS

- Use a proper IDE, PhpStorm or similar
- xDebug & Breakpoints
- Recursively Search Directories
- Use a local dev setup

LET'S CODE



<https://github.com/ChrisFlannagan/extending-woocommerce-example>

STAYING COMPATIBLE

USE CORE WOO FUNCTIONS

- Don't manipulate or get Woo data with \$wpdb
- WC() is pretty awesome
- Don't overstep WC_Product with custom Product wrapper class

CHANGE LOG IS YOUR FRIEND

- If project centric, don't update WooCommerce in production until you have completely QA'd it.
- Look for deprecated functions, then search your code
- Always update locally first with debug display and/or log on

FIXING BREAKS

- Woo is very good about inline documentation on deprecated functions and including the new method in the old one
- Woo templates change so your custom templates will need to as well.

THANKS!!

