

# Class Summary

- What is a Macro?
- The Basics
- Plugging Into WebGUI
- Using your Macro
- Demos
- Individual Project
- Group Project

# What is a Macro?

- Macros are WebGUI plug ins that allow dynamic content to be placed in static content.
  - Examples
    - Display data from a database on every page
    - Show the current date and time
    - Show which users are currently logged in

# The Basics

- Setup
  - Create a package in the WebGUI::Macro library
    - use WebGUI::Session
    - use WebGUI::Macro
- **process (params)**
  - The only required method
  - One parameter passed in
    - Used to generate a list of parameters
  - Returns a string

# Plugging Into WebGUI

- Add the Macro to the WebGUI config file

```
macro=# => High_SecId, \
/ => Plas_gatewayllr, \
0 => A_username, \
&IHICs => &OIHICs, \
&LROK => &OIROKs, \
&fmoTat => &fmoTat, \
```

- Restart Apache

# Using your Macro

- Add to your site with proper syntax

`^MacroName(value,"value",value);`

- Starts with a ^
- Has at least one character
- Ends with a ;
- Parameters are seperated by commas and contained withing parenthesis
  - Quotes are optional

# Demo

- Adding two numbers

```
package Web_UI::Macro::addNumbers;

use strict;
use Web_UI::Macro;
use Web_UI::Sections;

#-----
sub process {
    my $param = Web_UI::Macro::getParam($_[0]);
    my $number1 = $param{0};
    my $number2 = $param{1};
    my $sum = ($number1+$number2);
    return $sum;
}

1;
```

# Demo

- Commify a number

```
package WebGUI::Macro::Commify;

use strict;
use WebGUI::Macro;
use WebGUI::Session;
use WebGUI::Utility;

#-----
sub process {
    my ($param = WebGUI::Macro::getParams($_U));
    my $number = $param[0];
    my $comma_number = WebGUI::Utility::commify($number);
    return $comma_number;
}

1;
```

# Demo

- Count system users

```
package xcbUI::Xaccr::SystemUsers;

use 5.0100;
use Test::More;
use xcbUI::exceptions;
use Test::SQL;

-----
sub process {
    my $q = "select count(*) from users where status='Active'";
    my ($count) = Test::SQL->quickArray($q);
    return $count;
}

1;
```



# Demo

- Logged In Users

```
package UserGUI::Navbar::LoggedIn;

use strict;
use UserGUI::Navbar;
use UserGUI::Session;
use HTML::Table;

#-----
sub process {
    my $sql = "select a.username
              from users a, userSession b
              where b.userId <> 1 and a.userId=b.userId and a.expires > unix_timestamp()
              order by a.username";

    my @array = UserGUI::SQL->buildarray($sql);
    my $output = q|
        <table border="1" width="200" cellspacing="0" cellpadding="2">
            <thead class="tableHeader"><tr><th>username</th></tr>
            |;
    foreach my $user (@array) {
        $output .= qq| <tr><td class="tableData">$user</td></tr> |;
    }
    $output .= q| </table> |;

    return $output;
}

__END__
```