Extreme Forms

William McKee
william@knowmad.com
WebGUI Users Conference 2009
Forms are everywhere.
Forms make user interaction possible.
WebGUI makes forms easy!
Building a Form

1. Add a Data Form

2. Define Your Form

3. Add a Field

4. Define Your Field

5. Repeat!
How can I make better forms in WebGUI?

Start with good planning.

Use good markup.

Add useful interactive feedback.
Plan Your Form

Start with paper – draw out the form or use existing paper documents.

Be Brief – only require necessary fields. Consider your audience.

Prepare Responses – error messages, HTML reply, email notification.
Use good form markup.

Form Elements allow the user to enter information.

Labels describe fields.

Fieldsets group related fields.

Legends provide a title for fieldsets.

See Resources for template with proper form markup.
Interactive Feedback

Feedback helps your users!
Interactive Feedback

Character Counter

Inline Error Handling

Submit Confirmation

Email Address * Please enter a valid email address.

Please enter a valid email address.
JavaScript is your friend!

Each of the previous examples were implemented by adding JavaScript to a custom Form Template.

We use a combination of YUI & JQuery.
Why stop there?
How can I make extreme forms in WebGUI?

- Override tab form behavior
- Add dynamic fields
- Create custom form controls
Override Tab Form Behavior

List all entries. · Export tab delimited. · Add a field. · Add a Tab

ADD NEW TAB

Label

Subtext

What next? Add new Tab save
Override Tab Form Behavior

1. Select the Tab form template

2. Default display
Override Tab Form Behavior

Override Behavior

Event Details (optional)
If you want event coverage, please fill in these fields.

Name of Event
Purpose of event
Event Date
2009-09-04
Registration instructions

Order Now
Build Dynamic Fields

1. Select Location

2. Select a Physician from that location
Build Dynamic Fields

Implementation

1. Use YUI to add an onChange handler to Location
2. Submit a request to an article asset which contains a custom macro with a blank style template
3. Use the response to populate the <select> tag for physician list.
**Situation:** Our client needed an event management & registration Web site to support their annual volunteer event.

http://30days.leadershipcharlotte.org
Our Solution: We identified the following requirements:

• Administrative interface for managing projects.
• List of upcoming events on Homepage.
• Browseable list of all projects.
• Project details page.
• Online registration form.
Implementation: We decided that EMS was not a good fit for the basic registration needs of this project. We built an integrated solution using Thingy, SQL Reports and Dataform.

• Management interface: Thingy
• Upcoming events list: SQL Report
• Browsable project list: SQL Report
• Project details page: SQL Report
• Registration form: Dataform with custom template and custom form control
Projects are managed via the Thingy database.

The Registration form needed a way to select a Project from the list stored in the Thingy database.
Our custom control creates a select box using data stored in a Thingy table to be displayed in the DataForm asset (whew!)
Custom Form Controls

WebGUI has a very modular approach to form controls

All control logic is stored in the WebGUI::Form namespace

These modules are used by Data Form, Thingy, User Profile, etc.
Gotchas for using Custom Form Controls with the DataForm asset:

1) Module must be placed in the core WebGUI library path (e.g., /data/WebGUI/lib/WebGUI/Form)$^1$

2) The WebGUI::Asset::Wobject::DataForm asset must be modified to include the custom form control.

$^1$ As of WebGUI 7.7.19, this workaround is no longer necessary.
Other fun ideas:

Use auto-tab in fixed fields.

Pre-populate fields from user profile data.

Store user input with cookies.
Thank You!

William McKee
WebGUI Forums: knowmad
IRC: knowmad
Email: william@knowmad.com

Resources Available at
www.knowmad.com/wuc2009