Assigning Form Sections

- Assigning Sections to "Any"
- Assigning Sections to a Group
  - Assigning Multiple Sections to a Single Group
  - Assigning a Section to the Approving Group of a Prior Section
- Assigning Sections to a User
  - Assigning Multiple Sections to a Single User
  - Assigning a Section to the Approving User of a Prior Section
- Making Sections Optional
  - Optional Sections Based on a Field Having a Value
  - Optional Sections Based on a Field Not Matching a Value
  - Optional Sections Based on a Field Being Set

When creating a form you may want sections to be assigned to a given user or group based on user feedback, whether that user is the original submitter or some intermediate party routing the form’s future sections.

Assigning Sections to "Any"

Only the first section should be set to "Any." Subsequent sections set to "Any" must have a user or group assignment take place prior to the "Any" stage of the lifecycle. This can be an assignment based on user input or inherited from a previous section as described below.

If this assignment doesn’t occur nobody will get an email alert and nobody will see the form in their queue. In other words, that form will be "lost" or "stuck."

Assigning Sections to a Group

To assign a section to a group, create an empty <select> element in your HTML document. Set this element’s name attribute to "formcycle-assign-to-group-XXXXXX" where XXXXXX is the value of the id attribute of the form section you wish to assign. When this form is displayed to the user, the options available will be all of the candidate groups you specified as potentially having access to this section when Managing Form Section Permissions.

Example:

```html
<form id="Submitter" class="form-section visiblefromall">
  Approving Group: <select name="formcycle-assign-to-group-Approval_Group" class="required"></select>
  <div class="controls">
    <button type="submit" value="approve" name="submit">Submit</button>
  </div>
</form>

<form id="Approval_Group" class="form-section visiblefromall">
  Approver Name: <input type="text" name="name" class="required" />
  <div class="controls">
    <button type="submit" value="approve" name="submit">Approve</button>
  </div>
</form>
```

In this example the user of the "Submitter" section would be presented with the list of groups assigned to the "Approval_Group" section. The "Approval_Group" section would then only be editable by the group submitted by the original user.

Note: the text displayed to the user will be the Display Name of the group, not the group name you see when choosing permissions. This way group's can have meaningful internal names like "Weinberg Form Approvers" but appear as "Weinberg College" to the selecting user.

Assigning Multiple Sections to a Single Group

If this assignment doesn’t occur nobody will get an email alert and nobody will see the form in their queue. In other words, that form will be "lost" or "stuck."
Assigning multiple sections can be achieved by concatenating additional section ids separated by the pipe character. Example:

```
<select name="formcycle-assign-to-group-Initial_Approval|Final_Approval" class="required"></select>
```

When assigning multiple sections when the candidate groups set on the permissions screen don’t, the options available to the user will be the union of all the sections. So if the first section’s groups are “Faculty Affairs” and “Student Affairs” but the second section’s are “Student Affairs” and “Graduate Affairs,” the user would be presented with all three options (“Faculty Affairs,” “Student Affairs,” and “Graduate Affairs”). Group assignment overrides candidate groups defined in permissions. In this example if the submitter chose “Faculty Affairs” that group would be assigned both sections, even though “Faculty Affairs” wasn’t a candidate group in the second section.

Assigning a Section to the Approving Group of a Prior Section

To assign a section to a group that approved a prior section, on the future section add the class "formcycle-assign-to-group-of-XXXXXX" where XXXXXX is the id of the prior section the future section should inherit group permissions from.

For example, if the above form had subsequent sections that needed to be reviewed by the same group that did this initial approval at the end of the workflow, the final review section might look like this:

```
<form id="Final_Review" class="form-section visiblefromall formcycle-assign-to-group-of-Approval_Group">
  Approver Name: <input type="text" name="name" class="required" />
  <div class="controls">
    <button type="submit" value="approve" name="submit">Approve</button>
  </div>
</form>
```

Assigning Sections to a User

To assign a section to a user, create a text <input> element in your HTML document and set this element’s name attribute to "formcycle-assign-to-netid-XXXXXX" where XXXXXX is the value of the id attribute of the form section you wish to assign. Any NetID may be assigned to a section. If an invalid NetID is entered an error will be returned.

```
<form id="Submitter" class="form-section visiblefromall">
  Approving NetID: <input type="text" name="formcycle-assign-to-netid-Approver" class="required" />
  <div class="controls">
    <button type="submit" value="approve" name="submit">Submit</button>
  </div>
</form>
```

```
<form id="Approver" class="form-section visiblefromall">
  Approver Name: <input type="text" name="name" class="required" />
  <div class="controls">
    <button type="submit" value="approve" name="submit">Approve</button>
  </div>
</form>
```

There is currently a bug preventing this feature from working properly. If you need to assign multiple groups use “Assigning a Section to the Approving Group of a Prior Section” below.
User assignment always takes priority over group assignment, regardless of the order that the assignments take place in. For example, if the Approver section were assigned to a NetID and then later assigned to a group, it would still only be assigned to the earlier NetID.

User assignment also overrides permissions set to a given section. The NetID assigned to the section can edit the section if they otherwise wouldn’t have been able to access based on their group memberships.

**Assigning Multiple Sections to a Single User**

Assigning multiple sections can be achieved by concatenating additional section ids separated by the pipe character. Example:

```html
<input type="text" name="formcycle-assign-to-netid-Initial_Approval|Final_Approval" class="required"/>
```

**Assigning a Section to the Approving User of a Prior Section**

To assign a section to a user that approved a prior section, on the future section add the class "formcycle-assign-to-user-of-XXXXXX" where XXXXXX is the id of the prior section the future section should inherit user assignment from.

For example, if the above form had subsequent sections that needed to be reviewed by the same user that gave this initial approval at the end of the workflow, the final review section might look like this:

```html
<form id="Final_Review" class="form-section visiblefromall formcycle-assign-to-user-of-Approver">
  Approver Name: <input type="text" name="name" class="required" />
  <div class="controls">
    <button type="submit" value="approve" name="submit">Approve</button>
  </div>
</form>
```

**Making Sections Optional**

Sections that are made optional will be skipped unless the condition to activate those sections is met.

**Optional Sections Based on a Field Having a Value**

To make a section optional that is activated based on a field in a previous section having a particular value, give the optional section a class "formcycle-optional-onlyif-SECTION-var-VARIABLE-is-VALUE" where SECTION is the id of the form section that holds the variable you want to check, VARIABLE is the name of the field you want to check, and VALUE is the value of the field that should activate the optional section.

**Optional Sections Based on a Field Not Matching a Value**

To make a section optional that is activated based on a value in a previous section not matching a particular value give the optional section a class "formcycle-optional-onlyif-SECTION-var-VARIABLE-isnot-VALUE" where SECTION is the id of the form section that holds the variable you want to check, VARIABLE is the name of the field you want to check, and VALUE is the value of the field that causes the optional section to be skipped.

**Optional Sections Based on a Field Being Set**

To make a section optional that is activated based on a field in a being set to any value, give the optional section a class "formcycle-optional-onlyif-SECTION-var-VARIABLE-isset" where SECTION is the id of the form section that holds the variable you want to check and VARIABLE is the name of the field you want to check.