

KRAD - Configurable Rich Lightbox content - modal dialog & data framework

See [slide deck for KAI review 3-15-2012](#).

And [slide deck for Rice Collab team review 4-9-2012](#).

Purpose

The lightbox construct today is a modal dialog with a simple user-response set. The user must select a response, closing the lightbox, before they can interact again with the underlying page contents. These can be used for warning messages, for example, if proceeding with an action will cause data to be lost.

The lightbox construct in Rice 2.0 doesn't support more complex interactions with the user, with multiple controls and conditional logic, and rich styling. These are needed for good user experience and to better support a question framework.

This is related to the "Configurable, modal dialog box (needed for KNS equivalence)" requirement, which is the back-end work required to support the front-end experience described below. In KRAD today, we can have a lookup, inquiry, or direct inquiry in a lightbox, but we will also need the ability to have the richer type of content and interaction in a lightbox.

Detailed Description

The new rich lightbox construct should be a re-usable component that enables rich styling (from CSS stylesheets) and user interaction, to pose questions and collect the user's responses. It should support images, tables, declarative logic, checkboxes, radio buttons, dropdowns, and input fields (for example, for user to provide a text description, with details for their response choice), or keyboard navigation.

It should include the ability to progressively disclose questions and other UI choices depending on other selections made in the lightbox, to dynamically add to the content and grow in size (up to a max, scrolling supported thereafter) and to be configured to pass the data collected back to a resource, such as the application, form, or Quali Rules Management System.

Usage Scenarios

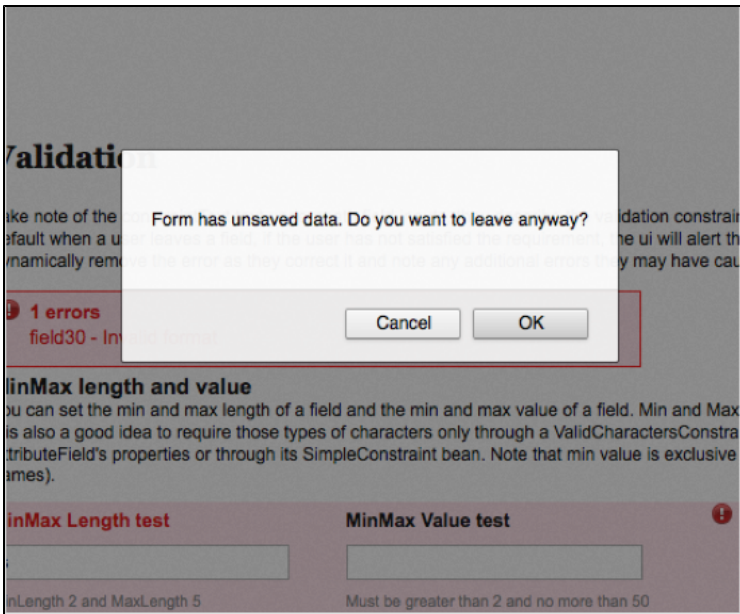
Usage scenario example 1: Faculty member selects to create a new course proposal - a lightbox appears asking him/her to choose whether to create it from a blank proposal template, from an existing course (copy over the content into the template), or from a course proposal already saved for another course. Based on this selection, additional content is displayed, enabling the faculty member to identify the course by code or title.

Usage scenario example 2: User decides to disapprove a document - a lightbox appears asking him/her to select the reason for the disapproval (for example, via a checkbox or radio button, depending on whether a single or multiple reasons are enabled). Based on the selection, additional content is displayed, for example, enabling the user to enter a short textual description to provide additional context (via an input field).

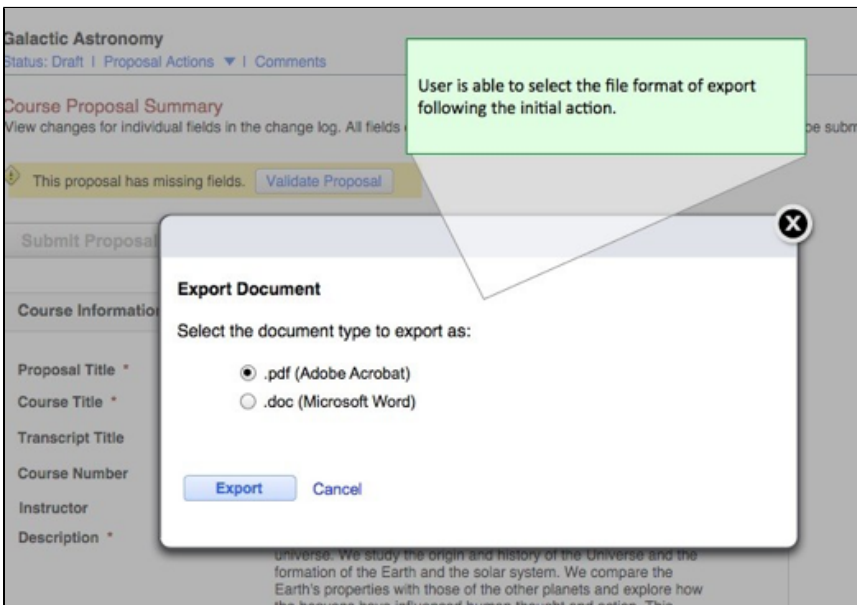
Usage scenario example 3: Faculty member selects to create a new research project and completes the first few fields in a pre-screening form in a lightbox. These questions determine which forms and workflow process will be needed for this project. The code/logic evaluates the answers to these questions and determines the content to add to the faculty member's research protocol definition and the workflow to follow (for example, will it need IRB review?). It adds the relevant tab/pages, for a multi-page form, and defines the submit action appropriately, to be evaluated by the Quali Rules Management System or to return to the calling application, or to enter an approval workflow at the appropriate node.

Mocks and Diagrams

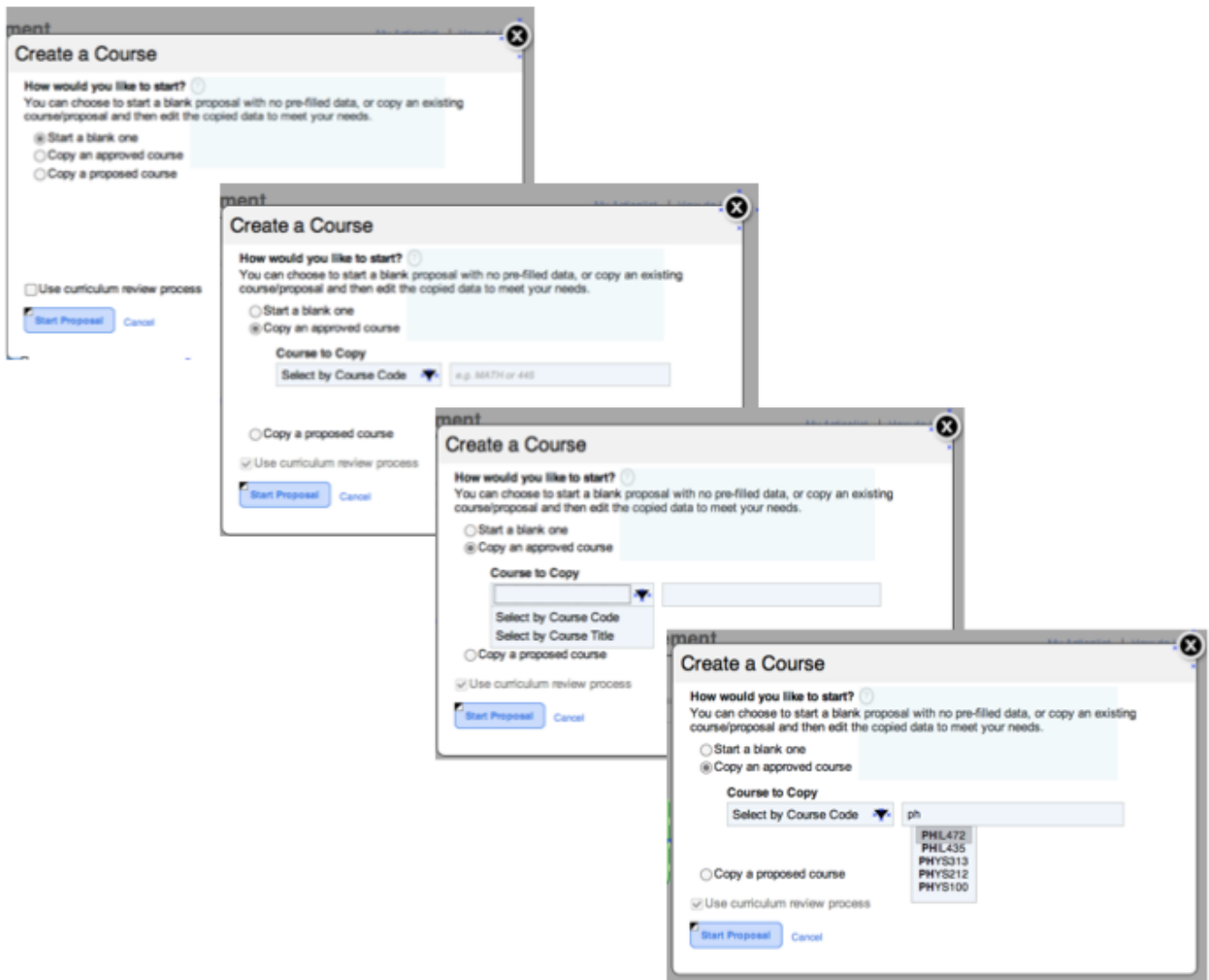
Pre-existing example 1 (basic example from the [KS User Interaction Model](#)):



Pre-existing example 2 (basic example from an older KS JIRA):



KS example 3, where progressive disclosure in the lightbox is needed (matches usage scenario example 1 above):



A Lookup Example 4, a lookup in a lightbox, progressively rendered lookup results are returned at bottom, as is typical behavior today with KNS lookups (that are not in a lightbox today):

Travel Account Lookup

Travel Account Number:

Account Name: *****

Travel Sub Account Number:

Sub Account Name: *****

Date Created:

Subsidized Percent:

Travel Fiscal Officer Id:

Travel Account Type Code:

Save Close Cancel

Example 5, Question framework - set of questions determine if a faculty member needs to complete additional forms to comply with policies for research projects (matches usage scenario example 3 above):

Create a Research Project

Which of the following are involved in this research?
Select all that apply:

- Human research participants
- Other live vertebrate animal participants
- Radioisotopes or radiation-producing devices
- Recombinant DNA

Create

Create a Research Project

Which of the following are involved in this research?
Select all that apply:

- Human research participants
- Other live vertebrate animal participants
- Radioisotopes or radiation-producing devices
- Recombinant DNA

Select DNA type(s):

- Transgenic plants or animals
- Human, animal or plant pathogens
- Biological toxins
- Experimental biological products for animals
- Field releases of plant pests or genetically-engineered organisms

Create **Cancel**

In this example, the lightbox progressively discloses additional questions if the "Recombinant DNA" checkbox is selected. The lightbox grows in size to accommodate this additional information. Also, the "question" architecture is able to pass the responses collected to the calling application (or to the Quali Rules Management System) to determine which forms will need to be completed for this project, and to pass into the forms any relevant data that has been collected. (Alternatively, the question framework itself could include those rules and associate them with this lightbox.)

Based on the user's selections in the lightbox example, there are 4 forms that must be completed for this research project:

Based on the results from this lightbox, the application could group the forms to present in a guided sequence for the researcher. Note that in this example, the IBC document itself could be pre-filled with the DNA types selected in the lightbox (and validation would make sure at least one type is selected if the "Recombinant DNA" checkbox is selected).

Step 1: IACUC

Step 2: IRB

Step 3: RSC

Step 4: IBC

Researcher is presented with all forms that must be filled out,
based on the responses within the lightbox construct.

First step: IACUC form input

Performance

If applicable, list expectations for performance (optimal and worst cases would be fine, give time in seconds):

References

Current JIRAs:

- <https://jira.kuali.org/browse/KULRICE-6675>
- <https://jira.kuali.org/browse/KULRICE-5372>
- <https://jira.kuali.org/browse/KULRICE-5276>
- <https://jira.kuali.org/browse/KULRICE-5984>
- <https://jira.kuali.org/browse/KULRICE-6931>

Example older JIRAs that show earlier KNS or other application-work for this Rice 2.2 KRAD requirement:

The rest of these below are older examples of where Kualu applications have needed this type of function:

- <https://jira.kuali.org/browse/KSLAB-1853>
- <https://jira.kuali.org/browse/KSLAB-1962>
- <https://jira.kuali.org/browse/KULRNE-3716>
- <https://jira.kuali.org/browse/KULRNE-3931>
- <https://jira.kuali.org/browse/KULRNE-3935>
- <https://jira.kuali.org/browse/KULRNE-4097>

Requirements Listing

List all requirements (individual verifiable statements) that indicate whether the work for this item has been complete. If there are requirements that are not essential to the functionality but would be nice to have if time allows, enter those under 'Secondary':

Primary:

Front-end primary requirements:

1. Currently, the KRAD 2.0 lightbox is in an iframe that only references KNS stylesheets. Priority 1 is the capability to reference any stylesheet (e.g., KS stylesheet) for the re-usable lightbox construct.
2. The rich re-usable lightbox construct must be able to include checkboxes, radio buttons, dropdowns, input fields (for example, for the user to provide a text description, with details for their response choice), images and tables.
3. The lightbox should allow for action buttons at the bottom of the lightbox for primary and secondary action fields.
 - a. The application should be able to define the action verbs displayed on the action fields.
 - b. The lightbox should be able to be cancelled (the standard cancel action should be placed at the bottom of the lightbox).
4. The lightbox should support the following keyboard behavior:
 - a. Upon opening the lightbox, KRAD should put focus on the first user-manipulable control within the lightbox (input field, checkbox group, radiogroup, etc.).
 - b. The Enter key should apply to the control that is in focus (the standard is that the Enter key and Space bar/key act as a mouse click - they activate the selected item, be that a link, button, or menu bar item to open a drop-down or pull-aside menu)
 - i. If this is not possible to achieve in this milestone (there is a separate [keyboard support requirement](#) that is not scheduled for this milestone), the Enter key should apply to the primary action button for the lightbox (not to the cancel button).
 - c. The user should be able to tab forward through each of the user-manipulable controls within the lightbox.
 - d. The ESC key should serve to cancel the lightbox.
5. The lightbox should have the standard close control (X) at the top right of the framework (styling tbd).
6. The application should be able to define the default size of the lightbox (either as a % of application window size, or as a fixed height/width).
7. For additional Bean/CSS styling capabilities and recommendations, see <https://docs.google.com/a/kuali.org/file/d/0Bwsymjqd8X3qT3ViMjluQ1g0SGM/edit> (Note, you may have to zoom in to read the words in some of the example mockups).

Back-end primary requirements

1. The lightbox framework must be able to include declarative, conditional logic, that is, to acquire or generate the HTML to display text and form elements to the user (for example, questions and their corresponding response choices in the above set of controls (see #2) - based on previous selections and if/then logic (progressive disclosure).
2. It must be able to pass back the responses to the calling action.
3. Creating custom questions should be configurable rather than requiring custom programming, for example, could be built in an xml description rather than in custom code logic. Applications should be able to choose from the following types of lightbox constructs provided by KRAD:
 - a. (we need to list here - include what is already in KNS)
4. KRAD should continue to support custom code logic. Applications should be able to create their own lightbox content, including question types as appropriate (beyond selecting from those provided by KRAD).
5. The user must be able to cancel as well as to complete all responses within a lightbox (support the standard cancel choice, no data saved).

Secondary:

Front-end secondary requirements:

1. It would be nice if the lightbox could auto-size itself to fit the content it is given to display (up to a max width/height determined by the application).
2. But it is not a primary requirement that the lightbox be able to dynamically re-size itself based on progressively-disclosed content. The additional scrollbars give users a way to access the additional content, if the content/size exceeds the current lightbox size. The aspect of dynamically-resizing up to a max size determined by the application, is secondary (a nice-to-have).
3. It would be nice if users could re-size the lightbox (stretch/shrink it).

Back-end secondary requirements:

1. The ability to save the responses in an incomplete lightbox.

Not included in this requirement, but related:

1. Note: What was not brought up by the applications in discussions at this time was a need to supply pre-canned question formats in a KRAD question framework, such as a Likert-scale object, with values pre-defined. Applications will be able to create these types of questions, for example, using a radiogroup control laid out horizontally or vertically in a lightbox, with values defined by the application.
2. The ability to navigate through all functionality in a lightbox via keyboard, in addition to via pointing (mouse, touchpad, etc.). *(A later Rice 2.2 KRAD requirement will determine the [keyboard support](#) in 2.2 - therefore, this work could be handled in the milestone assigned to that one, but the work done to fulfill the lightbox/question requirements should not make it more difficult to fulfill that one).*
3. The ability to navigate back/forward to previous states (lightboxes open or closed) through the browser controls. *(A later Rice 2.2 KRAD requirement will determine which states 2.2 will support through back/forward buttons -- whether the list will include lightbox states - #38 - [Back-button support](#) for js actions, pushing history of js actions to browsers history.)*

Dependencies

List any functional or technical work that must be completed before work on this item can begin:

Issues

List any issues that need to be resolved before work on this item can begin:

Functional:

1. item
2. item

Technical:

1. item
2. item

QA or Regression Testing Plan

List steps needed to test the basic functionality of this update, enhancement, bug fix

1. test/steps
2. test/steps

Checkoff

Functional Analysis Complete? **No (completed by SME)**

Needs Review by KAI? **No (completed by SME)**

Technical Analysis Complete? **No (completed by DM)**

Needs Review by KTI? **No (completed by DM)**

Estimate: **30 hours (completed by DM)**

Technical Design: [Link Here](#) **(completed by DM)**

Jira: [KULRICE-6675](#) **and** [KULRICE-7299](#)

Final Documentation: [Link Here](#) **(completed by DM)**

1. Added to QA: **No (completed by SME)]A**