

KRAD - Rich Messages

(Draft in progress below ... not completed)

Purpose

Support a rich message architecture, with the following characteristics:

1. Primary: messages can be styled, including capability to support line breaks, rich text (fonts, colors, and other stylistic aspects) and images
2. Primary: messages can include active links to other information (urls) or to other functions within the Kualu application or Rice infrastructure
3. Secondary: messages can include input fields (e.g. a dropdown choice embedded in a sentence).
4. Secondary: the message structure supports language and pronunciation tags, making it possible, for example, to display message syntax/variables in a different order and format, depending on the end users' language grammar (for example, right to left, left to right, and grammatical/syntax differences).

The following list of message types is a work-in-progress:

- text (message)
- instructional text
- constraint text
- checkbox items
- radio button items
- error messages (includes inline validation messages) - styling and links only
- hover help (tooltips) - styling and links only
- other help content (from help icon) - styling and links only
- headers (section titles, view header, page header, application header and footer content, banner content)

Detailed Description

Passive online help or textual messages, such as "dead end" error messages, are less helpful than active-dialog messages that, for example, enable a user to fix a problem in the context of doing their task. Applications that engage in these kinds of active dialogs with the user and whose help includes links to the appropriate targets within the application or elsewhere, rather than providing only textual information that requires the user to figure out and then navigate to other information in order to fix problems, are perceived as significantly easier to use than others, and lead to shorter task times (more efficient application use).

KRAD aims to enable Kualu applications to be able to design and support these types of richer messages. This document describes the requirements for the message structure and content types supported. There is a separate requirement document for the [Help Framework](#).

The technical architecture details including the message structure are being worked out now and will be added here.

Usage Scenarios

For scenarios related to online help, see the [Help Framework](#).

Usage Scenario One:

A student looking for a course types an invalid entry into a course ID field (there is no course with the value he/she entered). Instead of receiving an error message that states that the value is not valid or can't be found, the student receives an error dialog that includes the error message and also the question "Would you like to browse through course IDs now?" with a selectable action target for "Yes" (or is told "If you'd like to browse through course IDs now, select *" with the asterisk being an appropriate image - for example, it could be a lookup icon).

Usage Scenario Two:

A user doing a task in a Kualu application receives a cryptic error message with a message ID, because the server with the message text is temporarily unavailable or busy (performance times out). The message contains a link to display the message text and when the user selects the link, an independent browser window opens with content on the public website that contains the text for that message ID, and includes helpful information on how to resolve the problem. (Note, since this information is on the public web, the user can also use [Google translator](#) or [Bing translator](#) or any other publicly available web translator to translate the information, if it is not already in his/her native language.)

(add scenario where the message has entry fields, e.g. embedded in sentence)

Mocks and Diagrams

Include any mocks (for UI enhancements) or diagrams that might be helpful in understanding the issue:

Performance

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

References

Include links to other confluence pages or external resources that might be helpful for this issue:

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 'Non-Essential':

Primary:

1. From KRAD Req #11: Messages - preserve space using either or CDATA
2. item

Secondary:

1. item
2. item

Dependencies

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

1. item

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-6677](#)

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