KINETIC DATA
Your business. Your process.

KEG
Kinetic Enthusiasts Group

Kinetic Data Customer Conference & User Group

February 28 – March 1, 2013

REACH THE SUMMIT WITH KINETIC!

Multi-Part Requests/
Parent & Child Service Items

Agenda

- What is a multi-part request/cart?

- Child Service Item Components

- Parent Service Item Components

- Child Service Item Task Tree

- Parent Service Item Task Tree

- Questions

Who Am I?

- Anne Ramey

- Anne.Ramey@kineticdata.com

- 919.533.9770

- Consultant

- Community.kineticdata.com

What is a multi-part request (cart)?

Embedded Requests (Children)

Please complete each of the following Services before proceeding to submit this Order.

Each Service when completed will be indicated as being completed. Until all Services are complete this Order cannot be submitted.

| ATM | Desktop support | LAN Solution | Server Monitoring |

Service Details

* Details

Apply

Go to Review does not enable until all tabs have been completed

<< Return to Start   Save & Close   Go to Review >>

Child Service Item Components

- Each child must have an Attribute instance for an attribute created for this parent

- The "Type" field of the child is what determines what tab the child appears on in the cart page of the parent.

- A child must have a specific series of events on each page.

- A child must have a second "confirmation" page.

- The Submit Type field (field 700088475) will need to be added to your data set, if not already present.

- The child must use a different jsp for the Display Page (JSP) (from the advanced tab of the service item) that contains certain additional functions/features.

Cart Page Example

Child Service Item Forms/Resources

- **subFormPage.jsp::** This jsp is used as the display page for a child item. This contains the necessary functions and features to process as a child.

- **subFormPageReview.jsp::** This jsp is used as the review page for a child item. This contains the necessary functions and features to process as a child.

- **subform.css::** This CSS file contains the styles that ensure the child displays properly within the parent.

Child Service Item Elements

- **Section: Tabbed Details::** this removed (hidden) section is used to store data pulled over from the parent that isn't necessary for the user to view.

  - System Form:: contains the name/title of the child.

  - Primary Request ID:: stores the Instance ID of the parent Service Item. This question is mapped into the Originating ID field.

  - ParentKSR:: stores the request ID of the parent Service Item. This question is mapped to the Originating ID-Display field.

  - Submit Type:: This is set to Child if this is being called as a child subFormMarker:: contains SUBFORM if this is a child

  - _Tab_Mode_MarkAsComplete:: contains COMPLETED if the request is a child.

Child Service Item Events, Page 1

- **Set Primary ID::** Pulls primaryCsrv out of the URL

- **Hide/Remove for Child::** Used to hide elements filled in from the parent.

- **Get info from parent::** Sets a series of values from the parent. You MUST set the ParentKSR to the Customer Survey ID.

- **Tabbed Mode Deregister Form::** This on load custom action is just a clean-up for the parent.

- **Tabbed Mode set Defaults::** This on load set fields internal event sets some defaults used for processing of the child.

Child Service Item Events, Subsequent Pages

- **Set Parent Id::** This on load custom action sets the parent ID from page 1.

- **Remove buttons for child::** This on load insert-remove event hides custom buttons like "Cancel" and "Save & Exit" when the item is called as a Child.

- **Tabbed Mode::** This on load custom action ensures, now that we are past page 1, that the parent knows what the csrv of this in-progress child is, so the parent can correctly pull it up if the user returns to that page of children in the parent. It also hides the header and footer for the child.

Child Service Item Events, Last Question Page

- **Tabbed Mode::** This custom on load event will look different on the last page before the "Confirmation" page for the child. The "Submit" button on this page is probably called something like "Continue" or "Next" or "Submit" if the item is in stand alone mode. It should likely be named something more appropriate, ex. "Apply", if the item is a child. Note that this additional code is only necessary if the item is available as a stand alone. You can simply rename the page's submit button otherwise.

Child Service Item Events, Child Confirmation Page

- **ReviewMode::** This on load insert-remove event removes the only section on the page if the request is in review mode. This prevents this page from ever loading as part of a review

- **Remove Buttons::** This on load insert-remove event removes the submit button from view.

- **Process Variables::** This on load custom event sets the two questions on the page from the javascript on the page and calls the functions to remove the request from the parent's registry as an Incomplete item and to tell the parent that this child csrv is completed for this request.

Parent Service Item Components

- Questions, generally minimally requested by and requested for

- A cart and tabs where the user selects which services they wish to order

  - What services appear on this tab are controlled by a new attribute created for this purpose

- A services forms page that displays the children selected (in the cart), in iframes within the parent so they can be filled out within the parent

Cart Page Example

Tabs—populated based on Children "Type"s

Child listing with long description expanded.

Child listing with short description (default) shown. In this case it is the first 150 characters of the long description

Cart Page Elements

- Section: Cart Items:: contains the three text elements for the cart

- Section: Item Options:: contains the tabs for the services.
  - Tabbed Services:: contains a placeholder div used by the callback to display the services available for selection

- Section: System Access:: (hidden) contains the questions necessary for processing the cart.
  - Systems:: this question is used for storing the selected services.
  - Multi Form Indicator:: this question is used to specify that this item uses Multi-Form

Cart Page Events

- **Display Service Items::** This on-load, custom action does a simple data request for the attribute type created for this parent (in this example On-boarding) and uses either one of the tabbed Services callbacks to display the available services.

- **Remove for review request::** This event removes sections for the review request display.

- **valid Selection::** This before submit event checks that at least one selection has been done.

- **save Selection::** This before submit event saves the selected items into an answer for future use.

Cart Page Resources

- **tabbedservices.js::** Contains the functions that toggle the services into and out of the cart (both on click and on page load), the functions that toggle the long description on/off, and the function that saves the selected items into a question for future use.

- **tabbedservices.css::** Contains the CSS to make the item descriptions (short and long) display.

- **jquery-ui-1.8.12.custom.css::** Contains the CSS to make the tabbed pages display correctly

- **primo_icons/*::** These images are available for use on your tabbed services pages and can be referenced in your tabbedservices.css file for use/display

Cart Page Callbacks

- **TabbedServices.jsp::** Creates the tabs that hold the service item list for items with the attribute specifically created for use with this parent with selection box and short and long description (long description from the text of the attribute), categorized (tabs named) based on "Type" of the service items. The description can be shortened into the short description by grabbing the first however many characters, or, as in this case, everything before the first <br> in the description.

- **TabbedServicesReview.jsp::** Displays the status of the request instead of having items selectable.

Services Forms Page Example

Tab created for children selected on the previous page.

Please complete each of the following Services before proceeding to submit this Order.

Each Service when completed will be indicated as being completed. Until all Services are complete this Order cannot be submitted.

ATM ✓   **Firewall protection**

Service Details
* Details

Apply

Go to Review does not enable until all tabs have been completed

<< Return to Start   Save & Close   Go to Review >>

Body/Questions of the child

Services Forms Page Elements

- **Section: Hidden Details::** This (removed) section contains the hidden details that make this page function, both text elements and questions.

  - Systems Required:: contains the systems required table div

  - Systems Status:: contains the systems status table div

  - Systems Selected:: This question element contains the listing of systems that were selected on the previous page. It contains the events that create the tables in the above text elements.

  - Systems Selected Names:: This question contains the names of the services selected, for use in populating the tabs

  - Form IDs JSON:: This question is used to store the data from each child

Services Forms Page Elements, Cont.

- Section: Tab Section:: This section stores the tabs for the children items.

  - Info:: This text element contains some instruction as to how the tabs work

  - Tab Holder:: This text element contains the div tags for the tabs and the pages for those tabs where the children appear

  - Go to Review Message:: This text element is a reminder about the submit button not functioning until the tabs have been filled out.

Services Forms Page Events, On Page Load

- **Clear selection::** This on load set fields internal event makes sure we are looking at a clean slate to start off with

- **load Tabs In Review Mode::** This on load custom event builds the tabs for a review request.

- **Remove for review request::** This on load insert-remove event hides items for a review request

- **Trigger Load Systems Selected::** This on load custom event builds the tabs if this is not a review request. In this case you want to be sure you have the correct systems selected from the previous page.

Services Forms Events, On Question "Systems Selected"

- **Load Systems Selected::** gets the answer from the previous page's "Systems Selected" question and places it into "Systems Selected".

- **Re-Run load Selected Systems::** Triggers the LOAD to attempt again if the desired value was not found. These first two events basically loop until the answer is returned.

- **Load Systems Required::** Uses the callback loadSystemsRequired.jsp to populate a table in a removed (hidden) text element.

- **Load Systems Status::** Uses the callback loadSystemsStatus.jsp to populate a table in a removed (hidden) text element.

Services Forms Page Resources

- **multiQuest.js::** This javascript contains the functions that build the tabs, fill in the iframes within the tabs, store/maintain the status of the tabs, and handle the availability of the submit button.

- **multiQuest.css::** This CSS file contains styles for displaying the tabs of the service items.

- **tick.gif::** The complete tab image in this example

- **cross.gif::** The incomplete tab image in this example

- **subform.css::** This CSS file contains the styles for the sub-form (the child displayed inside the parent).
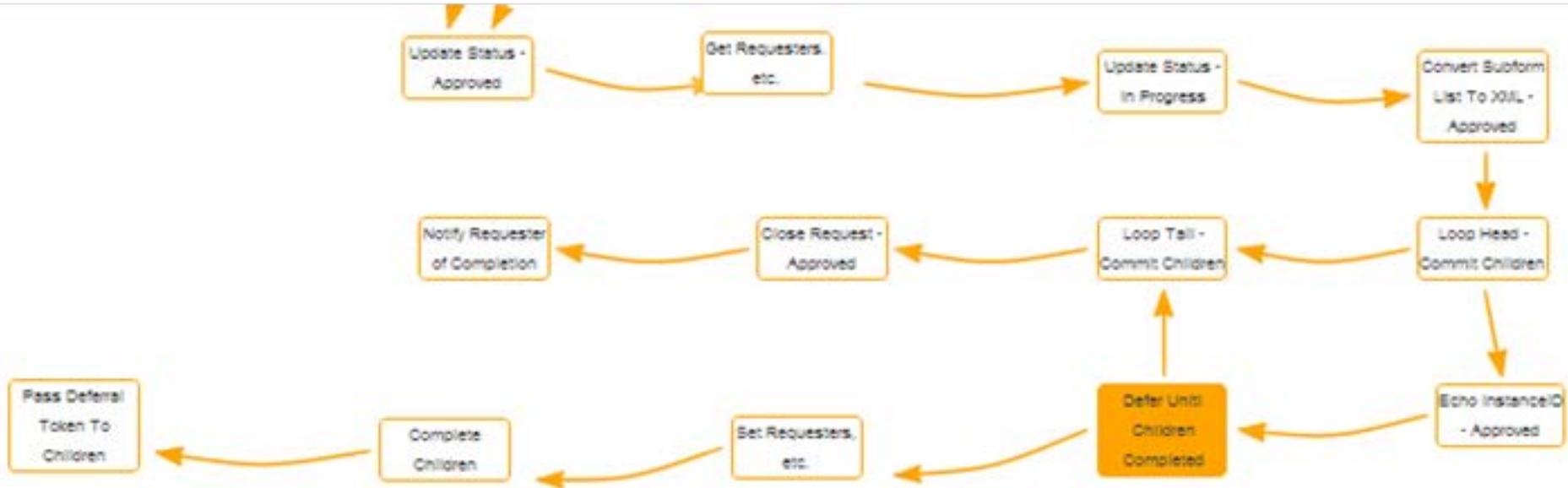
Services Forms Page Callbacks

- **loadSystemsRequired.jsp::** This callback creates a table of rows of instance IDs and Template Names for the items chosen on the cart page.

- **loadSystemsStatus.jsp::** This callback creates a table of rows of Name, Child Status, CSRV, Validation Status, Request Status for the children records created for the systems specified in the cart page. This allows us to reload the same children each time a user returns to this page.

Child Task Tree

- You must add a trigger node that has an action type of "Complete" and uses the deferral token you passed to the child in the parent tree.

- Can use submit type out of the data set on connectors to execute certain nodes only if the submission is or is not a child. Ex. <%=@dataset['Submit Type'] == "Child"%>

- Consider using the KS_SRV_Helper form to store/pass data.

Parent Task Tree

Questions?
(community.kineticdata.com)