An example of the functionality of the STD tool

Upon Double clicking the XML\_Parser executable, a file dialog box will open, asking for the “STD Folder”. This is a folder which should be contained inside the “icd” folder (current location is simulation/dev/common/icd/).

* This folder will contain sub folders, one for each model. The user must not select a sub folder inside this folder, otherwise the tool will only load this selected sub folder.
* Only the “std” folder must be selected

The ICDs tab

Once this folder has been selected, clicking open will open the user to the unsorted diagram screen.

* The left side of the screen contains a list off all loaded stdIcds. The user can click each entry and this icd will become the currently selected ICD.
* The right side of the screen contains a drawn data exchange diagram.

Guide to the diagrams

* Each rectangle at the top of the diagram is a Model. This models name is written inside the rectangle. The line leading down from this rectangle is this models line.
* Each horizontal line is a data transfer, with the red arrow showing the direction of data travel.
* The title at the top of the diagram can not be changed for the unsorted diagram, but can be changed for Functions and Tasks.

If I click “IcdModelAdu”. The program will read the stdIcd XML file, (not the stock ICD file) and draw a diagram, showing the Adu model as the left most rectangle in the diagram, and all proceeding models to the right.

* If the diagram can be navigated with scroll bars around the diagram scene, if the diagram is too large to fit in the diagram viewport.

Above the ICD list, are the 3 tabs used to navigate the different views the tool offers:

* ICDs (The current view)
* Functions
* Tasks

The Functions Tab

If the user clicks the Functions tab, a new view ill be visible to the user. This view is again separated into 2 sections

* The Top section is another diagram, this is the starting view for any new function the user will want to create. In this view, instead of the selected Model's rectangle being directly on the left, we have a new object, the “pilot” object. This object represents the pilots interaction within this function. Any way in which the pilot has input on the function, the data exchange arrow, just like in the unsorted diagram, will begin here.
* The lower section contains a box called Message Names. This box will contain and subscribed or published messaged that have to do with the currently selected ICD.
  + By clicking the arrow next to Pub (“published messages”), and expanding the selection, the user will see a list of all messages the ICD can currently publish, and expanding these selections further will show a list of all parameters included in these messages.
  + By clicking the arrow next to Sub (“Subscribed messages”), and expanding the selection, the user will see a list of all messages the ICD can currently receive, and expanding these selections further will show a list of all parameters included in these messages.
    - NOTE: If the ICD that did publishes these subscribed messages is not loaded at the start of the program, they will not be available to add to the diagram.

Creating a function

To create our first function, we should first name our diagram.

* To do this, click the “Diagram...” menu in the toolbar at the top of the window, and then click “Rename”. The user will then be shown a dialog box asking for the new name for the function. The dialog box will show the current title for the function, which will just be the name of the Icd we have selected. The function can be called anything the user wishes. Click Ok to confirm the new name.

Then we should add some messages.

* 'To add a message to the diagram, select it in the lower box, so that it is highlighted. The user can add both individual parameters, and whole Message blocks by selecting the item with the small triangle, that we clicked earlier to expand the selection.
* Once the selection is highlighted, clicking “Add Data Exchange” button will add the message to the diagram, in the lowest slot. With nothing in our diagram, this will be the only exchange currently in our diagram.
* The user can add as many messages as the like and the scroll bars next to the diagram allow navigation.
  + Note: Adding messages does not remove them from the message browser, allowing multiple instances of a message to be added to the diagram

Maybe we want to Edit and remove messages

* If we need to delete a message selecting the the data exchange by clicking on the name of it, will select that message. This is should put a border around the label, to signal it is selected.
* Once we have a selected message, right clicking the diagram will pop up a context menu that will have 6 options, the top 3 are the ones we need:
  + “Move Up” will move the selected message up once and move the one above this message down one.
  + “Move Down” does the same in reverse
  + “Delete” removes the message from the diagram.

If we need Pilot Interaction

* Clicking the pilot interaction button, allows us to add our pilot interaction to the function.
  + First a dialog box opens asking what our interaction is: This is what will show on the diagram
  + Then a 2nd diagram box open asking which part of the diagram the Pilot interacts with. Usually this will just be the selected ICD's rectangle, but allows for more options. This only lists any the selected icds published messages.
  + Removing and reordering pilot interactions is the same as all data exchanges, click to select the right click for the context menu.

If you need to add any comments to your function

* Function->Comments has a text box that allows multi line support for notes on changes or details you feel should be added. This text field is fully editable each time, so careful not to delete past entries.

To finish our function, we need to save it

* Click the “Function” menu in the top task bar, and then “Save Function”
  + If you have not renamed the function at this time, it will force you to rename the function before it will allow you to save the function.
* The next window will ask you to choose the file name. The directory it is showing is the ONLY place the file should be saved, which will be “STDs/\*\*ModelName\*\*/Functions”.

To Export as a JPEG

* The top taskbar, click “Diagram...” and then “Save to JPG”.
  + This will again open in the correct directory, and the user SHOULD NOT navigate away from this directory “STDs/\*\*ModelName\*\*/Diagrams”.

To open our function in the future

* Ensure the correct Icd is selected on the ICDs tab, before navigating to the Functions tab.
* Then select “Function” from the task bar, and then “Open Function”. This should open a file browser, in the correct directory, “STDs/\*\*ModelName\*\*/Functions”, and then simply double click on the function you want to open.

If at any point the Style of backgrounds and colors of labels

* This is the change validation, can be toggled my clicking “Functions” and “Toggle Style On/Off”
* If the spacing between models needs to be changed to allow longer or shorter data names, this can be changed in “Function”->”Spacing...”

Now we have a saved Function, we can move to the Tasks view, the third and final tab. Click the 3rd Tab, “Tasks” and this will show you a view similar to the ICDs view.

* The left hand list now shows every function that has been created for currently selected ICD, the program only check in “STDs/\*\*ModelName\*\*/Functions” for functions, hence why they MUST be saved in this directory.
* The right hand view shows a new diagram, with two circles.
  + The top skinny circle is our start point
  + The thicker bottom circle is our end point.

Adding a function is as simple as selecting it in the list and clicking “Add Function” (below the list).

Deleting and reordering functions is the same as in the functions view, select the added function so that you have a border around the label in the diagram, and then right click to view your context menu.

* The extra option on the tasks context menu allows you to quickly view the function you are dealing. Do not forget, the task you are working on has not been saved, selecting a new ICD from the first tab will lose all current work.

Styles can be toggle in much the same way, “Tasks”-> “Toggle Styles On/Off”.

Renaming Tasks is in the same place as renaming functions, but will rename your task as long as you are on that tab when selecting the action. Same goes for Saving to JPGs.

CHECKING FOR ICD CHANGES \*\*while buggy\*\* is implemented.

Clicking “XMLs” and “Check for ICD changes” Will check all stdIcd's against their ICD counterparts.

If a change is detected, a dialog box will let the user know that a message requires their input. It will tell the user the owner and name of the message, and ask whether the user wishes to:

* ADD: The programme will add the new message to the stdICD
* REMOVE: Ignore the new entry and not add it to the stdICD
* REPLACE: Selecting this option will offer a drop down list of all applicable messages that the new message could replace. Selecting an option from this list and clicking okay, will remove the message you have selected from the stdIcd, and add this new message in it's place.

These changes should be immediately visible in the stdIcd, however they will not be come visible in the tool until the user loads a function, or task that includes said function, that uses a varible that has been altered during this process.

When the user does load this function, the colour of the text label should change to reflect this

* Green for an Added message
* Red for a removed message
* Blue for a modified message (usually from a replaced process)

if the user selects this message and right clicks, the context menu that shows 6 options, the lower 3 allow the user to accept, reject and null the accepted status of this change. This should be reflected by the background colour of the label:

* Green for an Added message
* Red for a removed message
* Blue for a modified message (usually from a replaced process)

Any changes to the accepted status of chance s immediately saved to the stdIcd, NOT the function xml. Meaning the user is accepting or rejecting the change for ALL FUNCTION AND TASKS

**(Should probably be changed though)**