GUI – MESSAGE PATTERN – Action Progress

Revision history

2018-04-18	Robin Lamacraft	Original draft
2019-01-31	Rod Thompson	Separate from 05.87 GUI_Message Pattern Docs Add Headings per Specification template
2019-02-09	Rod Thompson	Replace window graphics (v9) LOOK AND FEEL – Add NOTE Add EXAMPLE MESSAGES section Add SCOPE section Add Issue Add Process Flowchart Add ACTIONS section Revise EVENT ACTIONS

This specification is one from the suite of Message Patterns used in HRE.

NOTE: for GUI Message Patterns which are a response to a status state identified within the HRE Client and HRE Server code, template text is required to be stored within the Substitution database tables. These will need to have an external tool built for the coders to preload that template into the HRE Project database. The method of achieving that preloading needs to be discussed and standardized with the Core Team.

Each “Message” GUI screen is modal (e.g. freezes other actions except for actions that are initiated from the “Message” GUI Screen) and full operation will be initiated according to the way the “Message” GUI screen is exited.

For each Message GUI screen:

- One or more text messages (these may be fixed translated text or they may have used substitution templates)
- One or more command buttons (these have language consistent labels) Command button order needs to be consistent with user progression from top to bottom of the screen. The number of command buttons can vary according to actual action being messaged
- A ‘Day-Time’ display showing local Day and Time. The format for this being a setting in the configuration of that screen
- The width and height can be changed to view the contents of the Template Text
- All GUI-Message patterns use BR_MessagePatterns to manage the access to data to be displayed and in the case of composing messages
- BR_MessagePatterns will use BR_Substitution to manage the Template Text fields.

GUI MESSAGE PATTERNS:

Nine separate Message patterns are used:

No:	Name	Window Heading	Code
1	Query message	Query	QY
2	Warning message	Warning	WG
3	Client Error message	Client Error	CE

4	Server Error message	Server Error	SE
5	Action Progress message	Action Progress	AP
6	Client Broadcast message	Client Broadcast	CB
7	Server Broadcast message	Server Broadcast	SB
8	Receive Inter-User message	Receive from Colleague	IR
9	Send Inter-User message	Send to Colleague	IS

The above list includes a two-character code, used to simply distinguish the patterns, when used in references from other specifications.

A reference from another specification (in the Required Services area) would (for example) have the following form:

GUI_Message (QY) or GUI_Message (SB)

REQUIRED SERVICES - Dependencies

First-Order Dependencies	Second-Order Dependencies (if not already listed)	Higher-Order Dependencies (if not already listed)
05.32 GUI_Select Configuration		
07.17 BR_WindowConfig	07.06 BR_GuiElement 07.02 BR_EntityLink 07.29 BR_Translation 07.16 BR_Message Patterns	
07.16 BR_MessagePatterns	07.02 BR_EntityLink 07.01 BR_EncodedString 07.11 BR_Logging Call Stack 07.24 BR_Setting GUI_Translation ???	
07.26 BR_Substitution	07.02 BR_EntityLink 07.29 BR_Translation	

SCOPE

This GUI module is called from other GUI modules to display the progress on specified actions.

Issue:
Does the Process Flowchart accurately reflect the intended design?

GENERAL

Typically, the use of this screen pattern is initiated when it is likely that an operation may take a longer time to complete. Some fields within this screen are periodically updated to indicate the progress of the operation.

The mockup included below is considered to be one possible way to illustrate operational progress. Use of the progress display element of this graphic is defined for use elsewhere in these specifications.

Alternative displays are considered worthy of further discussion.

Generally speaking, a consistent display approach should be used throughout HRE.

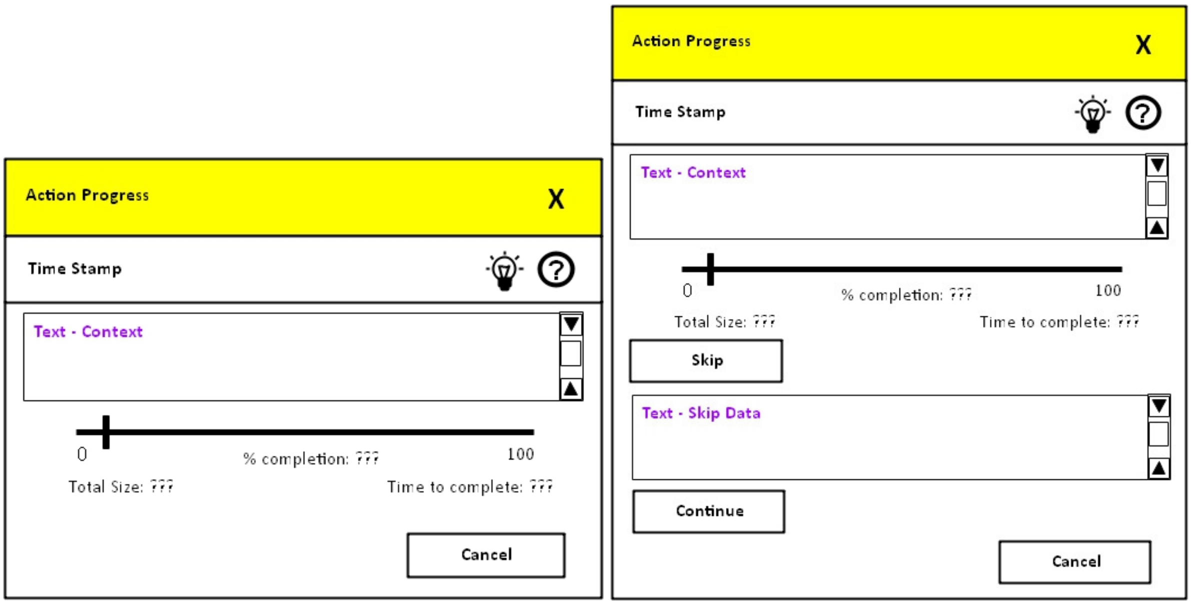
One exception is however considered to merit an alternative approach.
The strategy to load relatively static data from the database into memory at the 'Project Open' phase of use is anticipated to incur a time delay. A similar time delay may be expected in the 'Project Close' operation, as data in memory is written into database tables.

A proposal is to use this time to display a 'Credits List' of the coders who have contributed to the development of HRE. A suitable display design is required. The call for this display will come from each of the two modules defined.

SPECIFICATION

GUI ELEMENT USE	**ELEMENT TYPE**	**DESCRIPTION**
TOP BAR	HEADING TEXT	blank or translation of "Action Progress"
TOP BAR	RIGHT BUTTON	Only a "X" to "Close" the screen equivalent of "Cancel"
CONTEXT	TEMPLATE TEXT	Text describing the action being performed. This text may involve some substitution variables
STATUS BAR	PROGRESS BAR	Show proportion of task completed
STATUS NUMBERS	TEMPLATE TEXT	Total size, number processed, time to completion This text may involve some substitution variables
COMMAND 1	BUTTON	Typically "Skip" – only shown if a query. Placed above "Cancel". Action when clicked to signal to the requesting process to not process this item
SKIP DATA	TEMPLATE TEXT	Shown only processing skip error detected. Text describes the item with problem. This text may involve some substitution variables
COMMAND 2	BUTTON	Typically "Cancel" – aborts the operation reverting to prior state

LOOK AND FEEL



Progress Display Window - Mockups

NOTE: The first graphic shown above applies for normal use of this module.
The second graphic, involves an extension of the first, and is used when the Progress Display is associated with a Query.

Shape, Size, Color, Layout etc. determined in design. <u>On screen location</u> Initially inherited from the calling GUI window display Not User changeable	

ACTIONS

The fundamental operations are:

1. Open the window according to its saved window Layout (BR_WindowConfig)
2. Add the Time Stamp display
3. Load the Context Text display
4. Display the progress of the action

APPLICATION PROGRAMMING INTERFACE (API)

1. ????

LOGGING

Not applicable

EVENT ACTIONS

1) Keyboard actions
   NIL
2) Mouse actions
   <u>Left key</u>

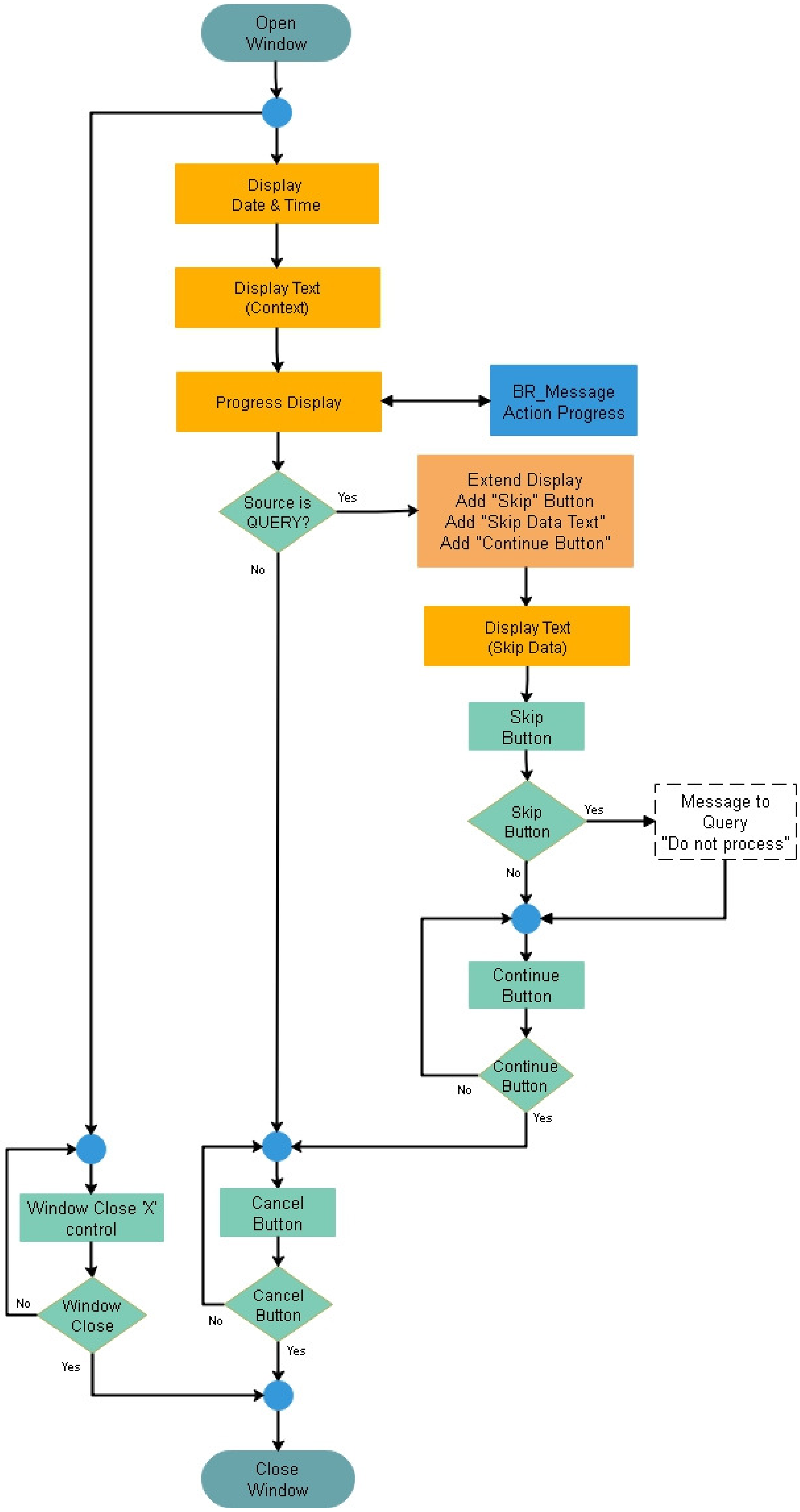
    a. Mouse hover over slider bar, hold key down and drag slider bar up or down
    b. Mouse key tap in slide bar box cause up or down movement.
    c. Mouse key click on each button, operates the button.

<u>Right key</u>

Action to be determined.

3) Icon actions
    a. left mouse key click on 'Reminder' icon:
    (opens 'Reminder' window for this 'Action Progress' Message display)
    b. left mouse click on 'Help ' icon:
    (opens the Help System, searches for the section on 'Action Progress Message')

PROCESS FLOW CHART



EXAMPLE MESSAGES (General Use)

Only the message element of the following table is displayed

Message	Context

```
ERROR: syntaxerror
OFFENDING COMMAND: --nostringval--

STACK:
```