HRE – SYNTAX FOR SUBSTITUTIONS - OVERVIEW

Revision history

2018-06-28	Robin Lamacraft	Original draft
2018-06-30	Robin Lamacraft	Considerable edits and extensions
2018-07-01	Don Ferguson	Edits; added Summary Marker Table at end
2018-07-01	Robin Lamacraft	Edits
2018-12-13	Robin Lamacraft	Revisions to include new cases, fix errors

SCOPE

HRE uses markup syntax in a template to identify what should be substituted for that markup symbol. This document provides a description of each type of construct. Some extensions may be added later.

PLEASE NOTE: These tables define the syntax and corresponding usage. However, when used by the user there must be a GUI including suitable toolbars that assist the user by in composing, validating and previewing example outcomes.

DATA RETRIEVAL SUBSTITUTION ELEMENTS

The constructs that retrieve data for substitution are as listed below.

SYNTAX	DESCRIPTION
[^*sample text***^]**	**ANY PLAIN TEXT FOR PARAMETER VALUES OR TO BE OUTPUT** All plain text is placed within"[^" "^]"delimiters NOTE: This includes leading and trailing blanks (when required).
[(*name***)]**	**REFERENCE TO A FOCUS SET**. Used as temporary storage
[()]	**REFERENCE TO THE CURRENT FOCUS SET**
>>[(*name***)]**	**SAVE FOCUS SET** as the current location
[=*name***=]**	**COMPLETE ALIAS** navigates and returns a formatted string, **[(***name of focus set***)] [=***name of alias***=]** e.g. **[(***person***)] [=***best.name***=]** NOTE: These do not support parameter lists
1. Normal use **[#***name1***#]** 2. Defining default parameters **[#***name2***(***param list***)#]** **3.** Using some defaults **[#***name3***(***param list***)** **<<[#** *name2***#]#]**	A **LINK ALIAS** takes an input object and changes the focus to another object which is linked to the input object. This provides the path to access properties of that linked object. These can be connected in a chain to access more data, for example **[(***person***)] [#***father***#] [#***father***#]** to access a person's father's father. This example shows how to create a new focus set **[(***person***)] [#***father***#] [#***father***>>[(***male.line.gfather***)]#]** NOTE: These link aliases support parameter lists. For LINK ALIASES these parameters control the navigation to, and selection of, target objects
1. Normal use **[$***name1***$]** 2. Defining default	An **OUTPUT ALIAS** takes an input object as the container of a collection of properties. Each output alias will retrieve a property, format it and return that string to the requesting module, for example

parameters **[$***name2***(***param list***)$]** 3. Using some defaults **[$***name3***(***param list***)** **<<[$***name2***$]$]**	**[(***person***)] [#***events***#] [$***birth.date***$]** to access a person's birth date. This example shows how to retrieve the person's father's birth date **[(***person***)] [#***father***#] [#***events***#] [$***birth.date***$]** NOTE: These output aliases support parameter lists. For OUTPUT ALIASES these parameters control the selection of properties and control how that value is formatted.

In many cases a substitution template will require the retrieval of clusters of data values that require the same preliminary database table query for a number of retrieved values. A focus set is an object that is temporary and stored in memory. Most often they only hold 1 value but they can store multiple values. Most templates will commence by storing the focus object's location from the API that made the request for the evaluation of a template. A Focus Set is an array of integer triplets (table number, record PID in that table and the object sub-type).

TEXT AND PAGE LAYOUT SUBSTITUTION ELEMENTS

These constructs modify the format and layout of the returned composed string. These elements either insert a code that represents a one-time action, or they insert an action which changes the style of the text or the page until that style is overridden by another action of the same type.

SYNTAX	DESCRIPTION
[:*name***:]**	The **[:***name***:]** form has 3 purposes and does not support parameter lists: 1. For special characters, inserts characters like a newline, tab, non-breaking space, etc 2. For page layouts, insert special actions like a start new page, etc 3. To be the named alias for a text or page layout or style definition that has preset parameter values of the form **[%***name***(***parameter list***)%]**
1. Defining default parameters **[%***name1***(***param list***)%]** 2. Using some defaults **[%***name2***(***param list***)** **<<[%***name1***%]%]**	The **[%***name***(***parameter list***)%]** form by use of its parameter values controls. It is used to define : 1. text styles by setting the properties of a font for that output until the next text style substitution 2. page layouts by setting the properties of a page layout until the next page style substitution. This controls page orientation, margins, headers, footers, paper size, etc The defined style is then aliased to a simple **[:***name***:]**

CONDITIONAL SUBSTITUTIONS

The comparison of a value with another similar value satisfies the test criterion. It can behave with a single object as focus or a subset as focus.

This construct has 5 parts in this order:

1. <u>The comparison test can have a number of different object types as the focus</u>
   NOTE: The type of the focus determines the list of legal functions for this construct.

a. **[?[(***focus***)][=***alias***=]** *other elements* **?]**

b. **[?[!***subset***!][=***alias***=]** *other elements* **?]**
PID comparisons between 2 subsets, to perform Boolean operations to create 2 new subsets by creating:
   - one subset containing the successful PIDs in the **[+ +]** segment
   - a second subset containing the failing PIDs in the **[- -]** segment.

2. The comparison function
   *function name***(***param list***)**
3. The TRUE result command content
   [+ *true commands***+]** If input is subset this can include **>>[!***subset2***!]**
4. The FALSE result command content
   [- *false commands***-]** If input is subset this can include **>>[!***subset3***!]**
5. The closer of the comparison
   ?]

Which becomes, in combination:

[?[(*focus***)] [=***alias***=]** *function***(***param list***)** **[+** *true commands***+] [-** *false commands***-]?]**

NOTES:

- One of the TRUE or FALSE content segments can be omitted
- This structure can be nested.

EXAMPLES:

All persons that were born before 1954

[?[!all**.**persons**!] [=**birth.year**=]** BEFORE**([^**1954**^])**
[+>>[!born.before.1954**!]+]**
?]

All women that were born before 1954

[?[!women**!]** ALSO_IN**([!**born.before.1954**!])**
[+ >>[!women.born.before.1954**!] +]**
[- >>[!not.in.both**!] -]** *(optional)*
?]

The functions used in the conditional substitution have 3 forms dependent on the function type:

1. Zero parameter – where a query is made about an object, is it set, is it redundant, etc
2. One parameter– where a property of the input object is compared with another value. These comparisons are often data type dependent. These are the most common type of function
3. Two parameters – where a property of the input object is compared against a value range, like a date being within a range or not.

Operations within the Subset Actions section (below) permit the access to and the saving of modified subsets. NOTE: The subsets created using these operations are discarded at the end of the evaluation of the Filter if their name commences with an underscore "_".

IS_SET TEST:

There is a special case that evaluates a clause in when one variable has a set value. There is no separate specification of a comparison function. It has the form:

[?[+*commands* **[***person***][=**birth.date**=]?***more camands***+]**
[- *other commands***-]** *(optional)*
?]

In this construct there is no test function and the **[+** .... **+]** clause is any set of commands. Within the **[+** ... **+]** clause there is one value reference that is succeeded by a question mark "**?**". The **[+** ... **+]** clause is only executed if that referred value is set. When "**?**"is used the value of the property is output in the returned string. When "**?~**"is used the value of the property is not output in the returned string
NOTE: This form allows other operations and values to be referenced within this form

In a special table **887 Substn Templates** where in the case of Name Style Output Templates all retrieved data values are related to one object type then the syntax above can be simplified to series of clause of the form

[+ [^some is set text**^][=**variable alias**=]? [^**more is settext**^]+]**
[- [^other non set text**^] [=**another variable alias**=] [^**other non set text**^]-]**
where the **[- ..... -]** clause is optional

This leads to the TMG template component

<some text **[**State**]**, **|** no state recorded, **>**

to become in HRE

[+[^some text**^][=**State**=]?[^**, **^]+] [-[^**no state recorded, **^]-]**
||||---------- ---------- ----- |||| |||| ----------||||

The HRE syntax enables a larger number of alternatives – (not shown here).

USER INPUT AND LOG OUTPUT

There are occasions where a template needs the user to input values to guide the execution. There are occasions where that user-entered value may need to be used in several locations within the template.

- To ask for a user input value from the Client window and save it:
  [?[^ *displayed question?***^]>>[~***variable name***~]?]**

- To use the user input and saved value:
  [~*variable name***~]**

NOTE: Run-time value input can be requested anywhere a comparison value or a parameter value may be entered.

There are occasions where the user needs to create a run-time progress log.

- To perform file operations like create, close, concatenate, clear, delete, etc
  [&*file action***(***param list***) >>[~***file path name***~]&]**

- To write records to a file
  [&= *any data output commands* **>>[~***file path name***~]&]**

PARAMETER LISTS

Parameter lists are used within many substitution elements.

A parameter list has the form of a comma-separated list of elements of the form:

(Parameter1 name **=** *Parameter1 Value***, ….. ,** ParameterN name **=** *ParameterN Value***)**

Where a parameter name:

1. can only be once in parameter list
2. can only consist of upper and lower case letters, digits and periods
3. must not use spaces within a name.

And where there are parameter values:

1. If the value is a number or a text string, it shall be delimited by [^ and ^]
2. If the value is a reference to an HRE retrievable value the value is HRE alias to that value.

Parameters can be omitted when the default value applies.

Each use of a parameter list, has

- An ordered list of the recognized parameters for that use
- A definition for each parameter for its value type and constraints
- A set of default values for each use of that parameter list.

Aliases for all places that use parameters are created as needed. This will mean that the templates that are used in most user-created templates will not show the inner detail, rather selection of an element or action will be by use of alias names.

NAMING AND REUSING BLOCKS OF COMMANDS

Often the same set of substitution commands are needed to be used more than once. There is syntax to define and then reuse defined blocks of code.

- To create and save a block of commands as a named entity for reuse:
  [<*block name***>]<<[<=** *any complete series of commands* **=>]**
- To reuse a block of commands:
  >>[<*block name***>]**

SENSITIVE DATA

In HRE there are 2 types of sensitive data:

- At the field entry level (where the whole field value is controlled), if:
  - the first 2 characters are **{{** then the data value is considered sensitive
  - the first 3 characters are **{{{** then the data value is never to be output

NOTE: This implies that all data retrieval must check for 2 **{{** at the start of a field value.

- Within a template or memo text
    - An inserted construct of the form
      [{ sensitive content **}]** is only output on specific request
    - An inserted construct of the form
      [{{ private content **}}]** is never to be output.

  NOTE:
    1. This implies that all data retrieval from a template must check for
       [{ ... **}]** or **[{{** ... **}}]** sequences in the content and operate correctly
    2. The sensitive content can contain any legal Substitutions for that context.

SUBSET NAMING - TEMPORARY and PERMINANT

In HRE a simple subset is a list of persistent identities (PIDs) of the same object type and object sub-type. The sorting of a series of columns uses property values associated with each PID to order the elements of the subset for display or subsequent computation.

NOTES:

- The type of object whose properties are being sorted is determined by retrieving properties of *subset1*
- Subset names that start with an underscore "_" are temporary e.g. "_AB", and will be destroyed when the filter has been completed
- Subset names that start with a letter are the equivalent of user created subsets and can be reused in another filter.

SORTING OF SUBSETS

In HRE a simple subset is a list of persistent identities (PIDs) of the same object type and object sub-type. The sorting of a series of columns uses property values associated with each PID to order the elements of the subset for display or subsequent computation.

For every property there must be a means of ranking their values, whether that is to taking in increasing (INC) or decreasing (DEC) values and finally whether to place the PID of a property that has no set value at the beginning (FIRST) or the end (LAST) of the ordering of that property, which is enabled by the 4 functions:

1. INC_FIRST
2. INC_LAST
3. DEC_FIRST
4. DEC_LAST.

This provides a means of ordering the properties used for sorting the subset and grouping their type of sorting to minimize the repetition in the specification.

[@[!*subset1***!]**
*sort function1***([=***prop1***=], [=***prop2***=]),.**
*sort function2***([=***prop2***=])**
>>[!*subset2***!]**
@]

In this case subset1 is used as input and the re-ordered subset is returned in subset2.

Use subset *women*, sort on *birth year* first, then *death year*, sort in ascending year order with unknown years placed at the beginning of the list:

[@ [!*women***!]** INC_FIRST **([=***birth.year***=], [=***death.year***=])>>[!***subset2***!]@]**

SUBSET ACTIONS

Using a subset created by the Subset Filter as input to a Subset Action can be used to modify data properties, like setting Flag values or find/replace for correcting spelling mistakes, etc. Because in HRE the Subset Action command can be inserted in a branch of a filter it is possible to perform more complex operations than those previously available.

These operate on the resultant subset of the previous operation unless the action is preceded by

>> [! *another subset name***!]**

To perform the equivalent of a TMG Utility data modification, the Subset Action function has the form

[?[!*subset1***!] [=***alias***=]** *function***(***parameter list***)**
[+@>> [!selected!]
[?[~*log***~]** EQUAL **([^**log**^])**
[+~*commands to output log record***~+]**
?]
[?([~*update***~]** EQUAL **([^**update**^])**
[+~*commands to make data change***~+]**
?]
+]
?]

This complexity should be hidden under the GUI.

MODIFYING THE DATABASE OR AUX FILES

To modify the HRE database or the HRE AUX files use the following syntax:

>> [(*focus***)][=***alias***=]**

NOTE: If the **[(***focus***)]** is a reference to a HRE AUX file the Table Number will be in a different range from the database tables.

SUMMARY TABLE OF SYNTAX MARKERS

MARKER	USE
[^*string***^]**	A text string
[‘*comment***’]**	A user comment
[{*sensitive***}]**	Content only to be output if sensitive data is to be output
[{{{*never output***}}}]**	Content never to be output in a report
[~*string name***~]**	Reference to a named string variable
[~?*script var***?~]**	Reference to named report script variable
[(*focus set name***)]**	Reference to a named focus set (can be **[()]** for the current focus set)
[=*complete alias***=]**	Reference to a named complete alias
[#*link alias***#]**	Reference to a named link alias (can have parameters)
[$***output alias***$]	Reference to a named output alias (can have parameters)
[:*text style***:]**	Reference to a named complete text style or page layout alias
[!*subset name***!]**	Reference to a named subset (can be **[!!]** for the current subset)
[.*Nth parameter***.]**	Reference to Nth parameter of requester
[%*text layout***%]**	Reference to a named detail text/page alias (can have parameters)
[<*use defined set***>]** **[<=***define set***=>]**	Reference to a named reusable sequence of commands Commands to define the contents of the block
[&*output file***&]**	A file output command
[?*test clause***?]** **[+***TRUE clause***+]** **[-***FALSE clause***-]**	A conditional clause Can include TRUE clause embedded after test within **[?**.....**?]** Can include FALSE clause embedded after test within **[?**.....**?]**
<<*use as input*	Use following object as source of default values
>>*save to name*	Save result in next referenced object
?[=*complete alias***=]**	? Prefix for data reference to test if value exists, output value
?~[=*complete alias***=]**	?~ Prefix for data reference to test if value exists, don’t output value