# HRE – RETRIEVED VALUE SUBSTITUTIONS - OVERVIEW

**Revision history**

| | | |
|---|---|---|
| 2018-06-18 | Robin Lamacraft | Original draft |

## SCOPE

HRE uses markup syntax in a template to identify what should be substituted for that markup symbol. There are many types of action that are available within the markup system. This document only describes the replacement of that markup symbol by a value that has to be retrieved and possibly formatted for use.

Retrieved value substitutions can be used in all types of substitution but their use may be restricted in Filters.

## OBJECT DATA VALUE IDENTITY

In HRE any object with properties is identified by a unique triplet of positive integers. These are:

- The object's table ID
- An object's sub-type ID
- A persistent record identity (PID).

And then field(s) that hold the value – operation on the fields are managed by specific code for each request.

SIMPLE EXAMPLE – retrieve the sex of a selected person:

1. The required sex data is in table **401 (BIOS)**
2. PERSON is Sub-type 1 in Generic Type BIOS in table **169 (ENTITY_SUB_TYPE_DEFNS)** value
3. The Selected PERSON's record PID is (say) 342
4. The PID of the table record in **877 (SUBSTN_DATA_ALIASES)** to access and process the Sex field value
5. In the API to retrieve this value, it is suggested that (401, 1, 342) are an integer array as the first parameter, and the integer PID of the Substitution Alias in table 877. This API should return a class with 2 elements, first an integer code for success, fail, etc and the second the retrieved and formatted value
6. The Substitution Alias may look like **~[=PERSON.SEX=]~**
7. This is what is called a COMPLETE ALIAS
8. Most probably this alias has been designed using a combination of modular components
9. The aim is to pre-define common value retrieval requests so users do not need to know about combining modular components to get most items.

## THE MODEL OF NAVIGATING AND RETRIEVING DATA VALUES

There are 2 types of value retrieval Alias:

- **COMPLETE ALIAS** for VALUE RETRIEVAL
  **[=alias name=]** – where the complete Alias specifies:
  - the starting point for retrieval
  - the navigation to other objects related to the start object
  - then the selection of the property to be retrieved
  - the output formal of the returned value.

  It's parameters were fixed when it was defined.

It is the form that is used in most templates.

- **PARTIAL ALIAS ELEMENTS**
  There are 3 modular element types used to compose a COMPLETE ALIAS
  - **HEAD ELEMENT**
    **~[=HEAD alias name(parameter list)#]~** – where this element specifies from where to start the retrieval process with optional parameter values provided and or default values for those parameters when parameters are omitted.

    There are a very small number (under 20) of possible HEAD ELEMENT types. Typically for every combination Object Type and Object Sub-Type there are a pair of HEAD ELEMENTS. Each type requires the Input Object Type, Input Object Sub Type and Input Object Record PID:
    1. OBJECT DATA RETRIEVAL – used for individual value retrieval
    2. SUBSET DATA RETRIEVAL – used for retrieval of values for subset members.

  - **LINK ELEMENT**
    **~[#a LINK alias name(parameter list)#]~** –this is an ***optional*** element that specifies where to navigate from the last HEAD or LINK ELEMENT to another object. It has optional parameter values provided and or default values for those parameters when parameters are omitted.

    The number of prior ELEMENT (HEAD or LINK) object types is a very small number. Most HEAD or LINK object types may be linked to new locations. There are a very small number (about 30) of possible LINK ELEMENT types. Typically for every combination Object Type and Object Sub-Type there are a pair of LINK ELEMENTS that require the Input Object Type, Input Object Sub Type and Input Object Record PID and the Output Object Type, Output Object Sub-Type:
    1. OBJECT DATA RETRIEVAL -
       returns a PID of the object navigated to
    2. SUBSET DATA RETRIEVAL -
       returns error code and Output Object Type, the Output Subset PID.

  - **OUTPUT ELEMENT**
    **~[#OUTPUT alias name(parameter list)=]~** – where this element specifies the property of the last input object (HEAD or LINK) and the format of that output using the optional parameter values provided and or default values for those parameters when parameters are omitted.

    For the prior ELEMENT (HEAD or LINK) object types there are as many properties that may be accessed as there are fields in the database for that final object type. Each final object type may have over 20 properties for that object type and sub-type combination that may be linked to that prior ELEMENT. Fortunately it will be listed in a separate list for each object type/object sub-type combination. They require Input Object Type, Input Object Sub Type and Input Object Record PID and the Output format parameters:
    1. OBJECT DATA RETRIEVAL -
       returns text string of the retrieved value
    2. SUBSET DATA RETRIEVAL -
       returns error code and an array of text of retrieved values.

- **FORMING COMPLETE ALIASES FROM PARTIAL ALIAS ELEMENTS**
    - A user can define a new COMPLETE ALIAS by combining PARTIAL ALIAS ELEMENTS using:
        1. 1 HEAD ELEMENT (with optional appropriate parameter values)
        2. *Optional* LINK ELEMENTS (with optional appropriate parameter values)
        3. 1 OUTPUT ELEMENT (with optional appropriate parameter values)
           using intervening "+" as:
           **~[=HEAD alias name#]~+~[#LINK alias name#]~+~[#OUTPUT alias name=]~**

NOTE: The 4 patterns of the delimiting "=" and "#" characters:

1. COMPLETE [=name=]
2. HEAD [=name#]
3. LINK [#name#]
4. OUTPUT [#name=]

## SOURCES OF RETRIEVED VALUES

There are 7 repositories of data that may be used with Substitutions
PDE – the currently open Project Database Environment (H2)

CCE – the Client Common Environment local file (JSON) Installation data and status
CUE– the Client User Environment file (JSON) User population properties and status
CPE – the Client Projects Environment file (JSON) Project properties and status of use

SCE – the Server Common Environment (JSON) Installation data and status
SUE– the Server User Environment file (JSON) User population properties and status
SPE – the Server Projects Environment file (JSON) Project properties and status of use

The content of these 3 Client and 3 Server files (JSON) is to be documented in the *Client and Server Auxiliary Files Overview*.

## SUBSTITUTION ALIASES

To make composition of templates much easier, HRE maintains a dictionary of named aliases grouped by operation type. The user can add their custom definitions to this collection. Once the user understands how to achieve some effect or access specific data, they can save the definition for reuse. That definition can be exported to other users. A user can import templates from other users. The user can supply a unique label and description for a substitution alias. The alias description will be displayed as the mouse hovers over the Alias name in the GUI.

Substitution Aliases are stored in table **877 (SUBSTN_DATA_ALIASES)** with one record for the start of each COMPLETE or PARTIAL ALIASES. There are a number of classifications of retrieval aliases:

- COMPLETE has no parameters
  All below have parameters and default parameter values
- HEAD_ONLY
- HEAD_LINK
- LINK_ONLY
- LINK_LINK
- LINK_OUTPUT_ONLY
- OUTPUT_ONLY

In table **879 (SUBSTN_STEPS)** a collection of records are used to define the actions for this Alias.