Published: 11/08/67

Identification

Macro_arg, Create_subst Control Commands
K. J. Martin

Purpose

Macro_arg and create_subst are macro control commands; that is, they are designed solely for use within macros. Each sets up a substitution of one character string for another. Thereafter, when the first of the character strings is encountered standing alone in any string in the macro, the second character string is substituted for it before the string is interpreted.

Macro_arg names a list of parameters of the macro. During macro expansion all freestanding occurrences of these parameters in the text are replaced by the arguments specified in the invocation of the macro. If the macro_arg control command appears more than once in a macro, each occurrence overrides the substitutions set up by previous occurrences.

The create_subst control command provides for the substitution of created unique character strings for specified character strings in the text of the macro. This facility is useful for avoiding naming conflicts when the name of a variable or segment (or whatever the character string represents) does not matter anyway. For example, a macro may need a temporary segment in a certain directory. At the time the user makes up his macro he doesn't know what entries may exist in that directory at a later date. If he tries to create a segment in that directory named "temp" he may run into a naming conflict. Therefore, he uses create_subst to substitute a created unique character string for the string "temp". The unique character string is distinct from other character strings created by the same method, and for all practical purposes will be unique.

Usage

macro_arg (string_1 string_2 ... string_N)

create_subst (string_1 string_2 ...string_N)

where the string_i are free-standing character strings in the text of the macro. A character string is considered to stand alone when it is delimited by ASCII characters that are not alphabetic, numeric, or the underscore(_).

In macro_arg if string_i is null then the corresponding substitution established by the previous macro_arg should not be removed.

It is expected that the number of arguments to the macro_arg command (parameters of the macro) is the same as the number of arguments that will be used when invoking the macro. Then the ith argument to the macro is substituted for string_i. If the number of arguments differ, the leftovers are essentially ignored. If there are more parameters than arguments to the macro, there are no substitutes to be made for the extra parameters. If there are fewer parameters, the extra arguments to the macro are not substituted anywhere (or their current substitution is left unchanged if another macro_arg has occurred previously).

Create_subst creates a unique character string by calling unique_chars (BY.15.01) and concatenating "crs_" to the front of the 15-character string returned. It then sets up the substitution of the created unique string for string_i.

Implementation

Macro_arg uses the segment macro_arguments prepared by the macro processor. As described in BX.18.01, a structure containing the arguments of the macro at hand has been allocated into the segment. That structure has the form:

```
dcl  1  args based (arg_ptr),

        2 previous_list ptr,

        2 n_args fixed bin(17),

        2 arg_array (arg_ptr->args.n_args),

          3 sub_index fixed bin(17),

          3 count fixed bin(17),

          3 chars char(100);
```

For each parameter, macro_arg takes the following steps:

1. If the parameter is a null character string, proceed to the next parameter.

2. Otherwise, obtain the index of the current substitution for the corresponding argument to the macro. This information is stored in arg_ptr->args.arg_array(i).sub_index which if 0, indicates that no substitution is currently in effect. If there is no substitution in effect, go to step 4).

3. Otherwise, destroy the current substitution by calling request_handler$delete_subst (see BY.4.01).

4. Install the substitution of the corresponding argument to the macro for this parameter by calling request_handler$install_subst (BY.4.01).

5. Store the index of that substitution (returned by request_handler$install_subst into arg_ptr->args.arg_array(i).sub_index.

If arg_ptr->args.n_args is less than the number of parameters, the extra parameters are ignored. If arg_ptr->args.n_args is greater, the extra arguments to the macro are not effected (the same effect as if the corresponding parameters had been null character strings).

Create_subst manipulates the substitution list through calls to the request handler (BY.4.01). For each element of its array of arguments, create_subst takes the following steps:

1. Call unique_chars (BY.15.01) to obtain a unique character string.

2. Concatenate "crs_" on the left of the unique character string.

3. Call request_handler$install_subst to add this substitution and remove any other substitution for string(i) at this level of the substitution list.