Introduction

Large variables for alphanumeric and binary data are based on the well known Natural formats A and B. The limitations of 253 for Format A and 126 for Format B are no longer in effect. The new size limit is 1 GB. These large static variables and fields are handled in the same manner as traditional alphanumeric and binary variables and fields with regard to definition, redefinition, value space allocation, conversions, referencing in statements, etc. All rules concerning alphanumeric and binary formats apply to these large formats.

In that the maximum size of large data structures (for example, pictures, sounds, videos) may not exactly be known at application development time, Natural additionally provides for the definition of alphanumeric and binary variables with the attribute DYNAMIC. The value space of variables which are defined with this attribute will be extended dynamically at execution time when it becomes necessary (for example, during an assignment operation: #picture1 := #picture2). This means that large binary and alphanumeric data structures may be processed in Natural without the need to define a limit at development time. The execution-time allocation of dynamic variables is of course subject to available memory restrictions. If the allocation of dynamic variables results in an insufficent memory condition being returned by the underlying operating system, the ON ERROR statement can be used to intercept this error condition; otherwise, an error message will be returned by Natural.
The Natural system variable *LENGTH can be used to obtain the number of bytes of the value space which are currently used for a given dynamic variable. Natural automatically sets *LENGTH to the length of the source operand during assignments in which the dynamic variable is involved. *LENGTH(field) therefore returns the size currently used for a dynamic Natural field or variable in bytes.

If the dynamic variable space is no longer needed, the REDUCE or RESIZE statements can be used to reduce the space used for the dynamic variable to zero (or any other desired size). If the upper limit of memory usage is known for a specific dynamic variable, the EXPAND statement can be used to set the space used for the dynamic variable to this specific size.

If a dynamic variable is to be initialized, the MOVE ALL UNTIL statement should be used for this purpose.

Note:
Due to performance considerations, the storage area that is allocated to hold the value of the dynamic variable may be larger than the value of *LENGTH. You should not rely on the storage that is allocated beyond the used length as indicated by *LENGTH, it may be released at any time, even if the respective dynamic variable is not accessed.

Back to Statement Usage Related Topics.

