

PROCESS GUI

Note:

This statement is only available under Windows.

```
PROCESS GUI ACTION action-name WITH { {operand1}
                                     {nX }...
                                     } [GIVING operand2]
                                     {PARAMETERSclause}
```

Operand	Possible Structure	Possible Formats	Referencing Permitted	Dynamic Definition
Operand1*	C S A	A N P I F B D T L G	yes	no
Operand2	S	N P I	yes	no

* The structure and format actually possible depend on the action to be performed.

Function

The PROCESS GUI statement is used to perform an action. An action in this context is a procedure frequently needed in event-driven applications.

For general information on these standard procedures, see the section **Event-Driven Programming Techniques** of the Natural User's Guide for Windows.

For information on the individual actions available, their parameters, and examples, see the section **Executing Standardized Procedures** of the Natural Dialog Components documentation for Windows.

action-name

As action-name, you specify the name of the action to be invoked.

Passing Parameters to the Action

As operand1, you specify the parameter(s) to be passed to the action. The parameters are passed in the sequence in which they are specified.

For the action "ADD", you can also pass parameters by name (instead of position); to do so, you use the *PARAMETERS-clause*:

PARAMETERS-clause

```
PARAMETERS {parameter-name = operand1}... END-PARAMETERS
```

This clause can only be used for the action "ADD", not for any other action.

If the action has optional parameters (i. e. parameters that need not to be specified), you can use the notation *nX* as a placeholder for *n* not specified parameters. Currently, the only actions that can have optional parameters are the methods and the parameterized properties of ActiveX controls.

nX

With the notation *nX* you can specify that the next *n* parameters are to be skipped (for example, 1X to skip the next parameter, or 3X to skip the next three parameters); this means that for the next *n* parameters no values are passed to the action. This is only possible for actions which are applied to ActiveX controls.

A parameter that is to be skipped must be defined as "optional" in the ActiveX control's method. If a parameter is defined as "optional", this means that a value can - but need not - be passed from the invoking object to such a parameter.

GIVING operand2

As *operand2*, you can specify a field to receive the response code from the invoked action after the action has been performed.