

PROCESS GUI

<pre>PROCESS GUI ACTION <i>action-name</i> WITH { { <i>operand1</i> } } [GIVING <i>operand2</i>] { <i>nX</i> } ... { <i>PARAMETERS-clause</i> }</pre>

This chapter covers the following topics:

- Function
- Syntax Description

Related Statement: OPEN DIALOG | CLOSE DIALOG | SEND EVENT

Belongs to Function Group: *Event-Driven Programming*

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 *Event-Driven Programming Techniques* (in the *Programming Guide*).

For information on the individual actions available, their parameters, and examples, see *PROCESS GUI Statement Actions* (in the *Dialog Component Reference*).

Syntax Description

Operand Definition Table:

Operand	Possible Structure	Possible Formats	Referencing Permitted	Dynamic Definition
<i>operand1</i> *	C S A	A U N P I F B D T L G	yes	no
<i>operand2</i>	S	N P I	yes	no

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

Syntax Element Description:

Syntax Element	Description
<i>action-name</i>	Action to Be Invoked: As <i>action-name</i> , you specify the name of the action to be invoked.
<i>operand1</i>	Passing Parameters to the Action: As <i>operand1</i> , you specify the parameter(s) to be passed to the action. The parameters are passed in the sequence in which they are specified.
<i>PARAMETERS</i>	See <i>Passing Parameters by Name</i> below.
<i>nX</i>	Parameters to be Skipped: With the notation <i>nX</i> you can specify that the next <i>n</i> 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 <i>n</i> 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 <i>operand2</i>	Field for Response Code: As <i>operand2</i> , you can specify a field to receive the response code from the invoked action after the action has been performed.

Passing Parameters by Name:

For the action ADD, you can also pass parameters by name (instead of position); to do so, you use the *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 (that is, 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.