Watchpoint Maintenance

A watchpoint causes the execution of a Natural object to be interrupted whenever the value of a variable changes. In addition, you can make the interruption dependent on a condition related to a specific variable value as described under Watchpoint Operators (see also *Set Watchpoint*) below.

The use of watchpoints allows you to detect unintended alterations of variables caused by objects that contain errors.

A variable is considered to have changed either when its current value differs from the value recorded when the watchpoint was last triggered or when it differs from the initial value. Comparative validation of watchpoint values is restricted to a field length of 253 bytes. For large variables that exceed the maximum length, only the first 253 bytes are used in the comparison.

A watchpoint is defined by specifying the name of the Natural object and the name of the appropriate variable.

The unique identifier for a watchpoint is the spy number assigned by the debugger.

Once a watchpoint has been specified, it remains set for the entire Natural session, unless you delete it.

To invoke the watchpoint maintenance function

- In the **Debug Main Menu**, enter function code `W`.

  Or:
  Enter the following direct command:

  ```
  WM
  ```

  The **Watchpoint Maintenance** menu appears.

This section describes the functions provided in the **Watchpoint Maintenance** menu and the fields and columns contained in a watchpoint screen.

- Set Test Mode ON/OFF
- Activate Watchpoint
- Deactivate Watchpoint
- Delete Watchpoint
- Display Watchpoint
- Modify Watchpoint
- Set Watchpoint

- Fields and Columns on Watchpoint Screens

Set Test Mode ON/OFF

See the section *Switch Test Mode On and Off*.

Activate Watchpoint

To set the current state of specified watchpoints to active

- In the **Watchpoint Maintenance** menu, enter function code `A`, an object name and/or a variable name.

  Or:
  Use the direct command `ACTIVATE`, the syntax of which is described in the section *Command Summary and Syntax*.

If you do not specify an object or a variable (or leave the default asterisk in the **Variable** field), *all* watchpoints are activated.

Deactivate Watchpoint

To set the current state of specified watchpoints to inactive

- In the **Watchpoint Maintenance** menu, enter function code `B`, an object name and/or a variable name.

  Or:
  Use the direct command `DEACTIVATE`, the syntax of which is described in the section *Command Summary and Syntax*.

If you do not specify an object name or a variable (or leave the default asterisk in the **Variable** field), *all* watchpoints are deactivated.

Delete Watchpoint

To delete specified watchpoints

- In the **Watchpoint Maintenance** menu, enter function code `C`, an object name and/or a variable name.

  Or:
  Use the direct command `DELETE`, the syntax of which is described in the section *Command Summary and Syntax*.

If you do not specify an object name or a variable (or leave the default asterisk in the **Variable** field), *all* watchpoints are deleted.

Display Watchpoint

To display a watchpoint

1. In the **Watchpoint Maintenance** menu, enter function code `D`, an object name and/or a variable name. If you do not enter an object name, the default object (if specified) is used.

   Or:
   Use the direct command `DISPLAY`, the syntax of which is described in the section *Command Summary and Syntax*.

   If a watchpoint has been set for the specified object and variable name, a **Display Watchpoint** screen with all watchpoint definitions appears similar to the example below:

```
10:25:32              ***** NATURAL TEST UTILITIES *****            2006-02-14
Test Mode ON                - Display Watchpoint -              Object

Spy number ..............  12
Initial state ........... active         Current state .. active
Watchpoint name ......... WATCHTEST1     DBID/FNR ....... 10/32
Object name ............. WATCHPGM       Library ........ SAG
Variable name ........... WATCHVARIABLE
Skips before execution ..     0          Format/length .. A 10
Max number executions ...     0          Persistent ..... N   Act.level ...  0
Number of activations ...     0
Error in definition ..... - none -

Commands ... BREAK









Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit  Last  Mod   Flip                    Alpha Hex   Canc
```

   The fields on the **Display Watchpoint** screen are described in *Fields and Columns on Watchpoint Screens*.

   If no unique watchpoint is found, the **List Watchpoints** screen (see below) appears.

2. On the **Display Watchpoint** screen, you can view the condition for watchpoint activation as specified with the watchpoint operator (see also *Watchpoint Operators*):

   Choose PF10 (Alpha) to display the operator and/or operand value in alphanumeric format. .

   Or:
   Choose PF11 (Hex) to display the operator and/or operand value in hexadecimal format.

   Choose PF22 (Cmds) to switch back to the default view of the **Display Watchpoint** screen, which contains the **Commands** field.

To list watchpoints

- In the **Watchpoint Maintenance** menu, enter function code `D`, an object name or a variable name. You can use asterisk (*) notation to specify a range of object names and/or variable names, for example, `ABC*`. If you enter an asterisk (*) only, all names are selected. If you do not enter an object name, the default object (if specified) is used.

  Or:
  Use the direct command `DISPLAY`, the syntax of which is described in the section *Command Summary and Syntax*.

  A **List Watchpoints** screen similar to the example below appears which lists all watchpoints set for the specified object(s) or variable name:

```
10:14:05              ***** NATURAL TEST UTILITIES *****            2006-02-14
Test Mode ON                   - List Watchpoints -              Object
                                                                    Top of data
Co No. WP Name      Library  Object    DBID   FNR Stat Skips Execs Count P E
       *___________ *_______ *_______             I  C
       *_____________________________________________________________________
__    1 NAME        SAG      DEBPGM      10    32 A  A     0     0     0 N
        EMPLOYEES-VIEW.NAME
__    5 #MAKE       SAG      DEBPGM      10    32 A  A     0     0     0 N
        #MAKE
__   10 LEAVE-DUE   SAG      DEBPGM      10    32 A  A     0     0     0 N
        EMPLOYEES-VIEW.LEAVE-DUE
__   11 WATCHTEST2  SAG      DEBPGM      10    32 A  A     0     0     0 N
        TESTWP
__   12 WATCHTEST1  SAG      WATCHPGM    10    32 A  A     0     0     0 N
        WATCHVARIABLE
__   13 WATCHTEST3  SAG      DEBPGM      10    32 A  A     0     0     0 N
        WPTEST



Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help        Exit  Last        Flip  -     +                       Canc
```

  The list is sorted in ascending order by the spy numbers contained in the **No.** column.

  For details on the columns contained in the **List Watchpoints** screen and the line commands that can be executed on any list item, refer to *Fields and Columns on Watchpoints Screens*.

Modify Watchpoint

To modify a watchpoint

1. In the **Watchpoint Maintenance** menu, enter function code `M`, an object name and a variable name. If you do not enter an object name, the default object (if specified) is used.

   Or:
   Use the direct command `MODIFY`, the syntax of which is described in the section *Command Summary and Syntax*.

If a unique watchpoint has been specified, the **Modify Watchpoint** screen appears where you can change field entries. The fields on the **Modify Watchpoint** screen are described in *Fields and Columns on Watchpoint Screens*.

If no unique watchpoint is found, the **List Watchpoints** screen (see *Display Watchpoint*) appears.

2. On the **Modify Watchpoint** screen, you can change the condition for watchpoint activation as specified with the watchpoint operator (see also *Watchpoint Operators*):

   Choose PF10 (Alpha) to modify the operator and/or operand value in alphanumeric format. .

   Or:
   Choose PF11 (Hex) to modify the operator and/or operand value in hexadecimal format.

   Choose PF22 (Cmds) to switch back to the default view of the **Modify Watchpoint** screen, which contains the **Commands** field.

3. When you have finished editing the watchpoint definitions, choose PF3 (Exit) or PF5 (Save) to save any modification. If you choose PF12 (Canc), the watchpoint remains unchanged.

Set Watchpoint

To add a watchpoint for a session

- In the **Watchpoint Maintenance** menu, enter function code S, an object name and a variable name.

  Or:
  Use the direct command SET, the syntax of which is described in the section *Command Summary and Syntax*.

  Or:
  Before executing a Natural object:

  - Invoke the **List Object Source** screen (see *List Object Source*).
  - In the **Source** column, position the cursor at a variable name and choose PF18 (Se Wp).

  If you specify not an object name but a valid variable name, the name of the default object (see the section *Start the Debugger*) is assumed. If no default object is specified, a selection window appears that displays all objects available in the current library. If no default object is specified, a selection window appears that displays all objects available in the current library.

  If object name and variable names are specified correctly, the watchpoint is set immediately and a corresponding confirmation message is displayed on the screen. A watchpoint set for a dynamic variable or an X-array is only validated during program execution. See also *Maintenance and Validation* for information on validity checks of debug entries.

  The watchpoint receives the default command (BREAK), its initial and current state are set to active and no execution restrictions are specified. Note that if you delete the default command BREAK when setting a watchpoint and you do not enter any command that issues a dialog, there is no way for the debugger to receive control during program interruption.

This section covers the following topics:

- Watchpoint Operators

Watchpoint Operators

You can specify a condition for watchpoint activation by entering an operator and an appropriate operand (if relevant) on a watchpoint maintenance screen.

To specify watchpoint operators

1. On the **Set Watchpoint** or **Modify Watchpoint** screen of the selected watchpoint, choose PF10 (Alpha) if you want to specify an operator operand in alphanumeric format.

   Or:
   On the **Set Watchpoint** or **Modify Watchpoint** screen of the selected watchpoint, choose PF11 (Hex) if you want to specify an operator operand in hexadecimal format.

   Two input fields appear in the lower half of the screen.

2. In the left input field, enter one of the watchpoint operators listed in the following table.

   In the right input field, if relevant, enter the operand value to be compared with the variable. For watchpoints with operators specified for dynamic variables (alphanumeric or binary), the operand values will be compared from left to right. Since the field length of a dynamic variable varies, up to 253 bytes can be entered as comparative value. If the current length of the dynamic variable is shorter than the maximum comparative length of 253 bytes, the comparison is made only in the current length of the dynamic variable.

Operator	Explanation
MOD	Modification. Activates the watchpoint each time a modification of the variable occurs. This is the default setting.
EQ	Equal to. Activates the watchpoint when the variable has been modified and when the current value of the variable is equal to the specified operand value.
NE	Not equal to. Activates the watchpoint when the variable has been modified and when the current value of the variable is not equal to the specified operand value.
GT	Greater than. Activates the watchpoint when the variable has been modified and when the current value of the variable is greater than the specified operand value.
GE	Greater than or equal to. Activates the watchpoint when the variable has been modified and when the current value of the variable is greater than or equal to the specified operand value.
LT	Less than. Activates the watchpoint when the variable has been modified and when the current value of the variable is less than the specified operand value.
LE	Less than or equal to. Activates the watchpoint when the variable has been modified and when the current value of the variable is less than or equal to the specified operand value.
INV	Invalid contents. Activates the watchpoint each time the value assigned to a variable of the Type N, P, D or T does not comply with the following conditions: N Numeric unpacked. P Packed numeric. D Date range from 1582-01-01 to 2700-12-31. T Time range from 1582-01-01 00:00:00.0 to 2700-12-31 23:59:59.9.

You can choose PF22 (Cmds) to switch back to the default view of the **Set Watchpoint** or **Modify Watchpoint** screen, which contains the **Commands** input field.

3. Choose PF5 (Save) to save the operator definitions.

   Or:
   Choose PF12 (Canc), to leave the operator definitions unchanged and exit the **Modify Watchpoint** screen.

Fields and Columns on Watchpoint Screens

The fields contained in a **Display Watchpoint** or a **Modify Watchpoint** screen and the columns of a **List Watchpoints** screen are described in the following table:

Field	Column	Explanation
Test Mode		Indicates whether test mode is set to `ON` or `OFF`.
Object		Displays the name of the default object (see *Start the Debugger*) if specified.
	Co	Input field for any of the following line commands: `AC` Activate watchpoint `DA` Deactivate watchpoint `DI` Display watchpoint `MO` Modify watchpoint `DE` Delete watchpoint `?` List valid line commands `.` Exit watchpoint screen
Spy number	**No.**	A unique number assigned by the debugger when setting the watchpoint.
Initial state	**Stat I**	Specifies the initial state and the current state of the watchpoint: active (`A`) or inactive (`I`).
Current state	**Stat C**	
Watchpoint name	**WP Name**	The name of the watchpoint. The default name for a watchpoint is the name of the variable concerned. Valid values: 1 to 12 characters. Names that exceed the field size will be truncated after 12 characters. On the **List Watchpoints** screen, the watchpoint name is listed in the first line, above the variable name.
DBID/FNR	**DBID**	The database ID (DBID) and file number (FNR) of the system file where the Natural object is stored.
	FNR	
Library	**Library**	The name of the library that contains the object.
Object name	**Object**	The name of the object available in the current library or one of its steplibs. If you want to specify a system variable as a watchpoint, enter an asterisk (*) in the **Object name** field.

Field	Column	Explanation
Variable name		The name of a user-defined, global or system variable. If the variable is part of a group, it may be prefixed by the group name. If you want to specify a system variable, enter an asterisk (*) in the **Object name** field. For an array, an index description has to be specified (watchpoints can be defined for single elements only). On the **List Watchpoints** screen, the variable name is listed in the second line, below the watchpoint name. See also *Variable Maintenance* for further details.
Skips before execution	**Skips**	Determines that the watchpoint is not to be executed until the condition set for the watchpoint has been fulfilled (see also Watchpoint Operators). Valid values: 0 (default) to 32767.
Max number executions	**Execs**	Any value greater than zero (0) determines the maximum number of watchpoint executions. Valid values: 0 (default) to 32767.
Number of activations	**Count**	Indicates how many times the watchpoint condition for the variable was met as specified with the watchpoint operator. The counter is reset when a program is started at Level 1.
Format/length		The Natural data format and length of the variable, for example, A10.
Persistent	**P**	Marks a watchpoint as persistent. Persistent watchpoints are not restricted to the Natural object for which they are defined, but apply additionally to all subordinate program levels. Persistent watchpoints only make sense for variables that are passed to a subprogram by reference and not `BY VALUE RESULT`: see the relevant parameter description of the `CALLNAT` statement in *Parameters - operand2*, in the *Statements* documentation. **Restriction:** Persistent watchpoints are not allowed for variables defined in a parameter or context clause. Valid value: `Y` (Yes) or `N` (No). `N` is the default.
Act. level		Refers to Persistent. Indicates the program level at which a persistent watchpoint was activated automatically.

Field	Column	Explanation
Error in definition	**E**	Indicates an invalid watchpoint definition. This error may occur if the executing program is recataloged during debugging after the respective variable definition was modified. A watchpoint set for a dynamic variable or an X-array (eXtensible array) is only validated during program execution.
Commands		Up to six debug commands. Enter one command per line. For a summary of all available commands, see *Command Summary and Syntax*. **Caution:** If you delete the command BREAK and you do not enter any command that issues a dialog, there is no way for the debugger to receive control during program interruption.