

HI_RefreshData

%HI_RefreshData function

Function

The function **%HI_RefreshData** ([Control functions](#)) may be used:

- for display of [Browser](#) type - it updates displayed data. It also allows to change the selection condition or sorting of data.
- for display of [Report](#) type (containing the report of [Crystal report](#) type) - it sets the values of parameters, for which the function [%HI_ReportPar](#) was called.

Declaration

```
%HI_RefreshData(
    INT in refId
    [, TEXT in whereStr
    [, TEXT in orderByStr
    [, in par1[, in par2[, ...]]]]]
)
```

```
BOOL %HI_RefreshDataSync(
    INT in refId
    [, TEXT in whereStr
    [, TEXT in orderByStr
    [, in par1[, in par2[, ...]]]]]
)
```

```
%HI_RefreshDataTrans(
    INT in refId
    , INT in transHandle
    [, TEXT in whereStr
    [, TEXT in orderByStr
    [, in par1[, in par2[, ...]]]]]
)
```

Parameters

refId	Reference to graphic object (reference variable).
transHandle	A unique number (handle) of the connection to a database obtained by calling the DB_TR ANS_OPEN action.
whereStr	SQL condition WHERE for selection limit of displayed data (for Browser type only). Entering the condition <code>#empty#</code> doesn't read any data into the Browser
orderByStr	SQL command ORDER for selection order control (sorting) (for Browser type only)
par1, par2, ...	List of constants or local variables that will be used instead of the parameters of parameterized SQL condition.

Note

Condition and sorting can be also defined when you are configuring the graphic object in the process. These will be changed when the function **%HI_RefreshData** is called and displaying is updated. These parameters are not used for display of [Report](#) type.

The condition that has been changed by calling the function **%HI_RefreshData** may be [parameterized](#). The parameterization makes the work with SQL database easy because the preparing (compilation) of parameterized SQL expression can be done only when changing the condition or sorting (i.e. text format of SQL expression). The proper setting of database parameters (e.g. Oracle: `session_cached_cursors`) enables to recycling the cursors (the compiled commands) between the callings of **%HI_RefreshData**.

The rules for the parameters:

- parameter **refId** must be always valid,
- rules for **whereStr** and **orderByStr**:

	Entered valid	Empty or entered but invalid
whereStr	change of condition	The last entered condition is valid (if it has not been entered, there is an empty string).

orderB yStr	change of sorting	The last entered sorting is valid (if it has not been entered, there is an empty string).
------------------------	-------------------	---

- if **par1** to **parN** are entered, their quantity must correspond to the condition (count #PAR#), however when changing the condition, they may not be omitted,
- if any of parameters from the range **par1** to **parN** is invalid, it means NULL.

The function **%HI_RefreshData** makes its action in the way that new conditions of display delivers to proper displayer and ends itself aside from a result or success of display.

The function **%HI_RefreshDataSync** waits on the selection according to new parameters and informs about the success of action by its value (@TRUE = successful execution, @FALSE = error). If the browser, which displays DB table, is configured so that it is initiated automatically (the parameter "Initial query" is ticked off), the function cannot be used in the initiate (BEGIN..END) part of the picture. In such case it returns @FALSE.

The **%HI_RefreshDataTrans** function, unlike the **%HI_RefreshData** function, only schedules a data refresh of the displayer. The refresh is performed only when the **transHandle** transaction is committed with the **DB_TRANS_COMMIT** action. If the database operation is canceled (action **DB_TRANS_ROLLBACK**), data refresh will not be performed. If the **%HI_RefreshDataTrans** function is called multiple times for the same displayer, only one refresh is performed after the commit (with the parameters of the last call to the **%HI_RefreshDataTrans** function).

Note: The **transHandle** transaction can also be opened against another object of the **Database** type (other than the **Database**, which is the parent of the table whose data is in the displayer), but it is necessary that this object of the **Database** type has the same parent (the DBManager process) as the **Database**, which is the parent of the table whose data is in the displayer.

If a picture with a displayer of **Browser** type has been resized and the displayer has dynamic number of lines depending on browser's height configured (see the parameter **Lines**), the function **%HI_RefreshData** must be called to refresh the current page with a changed number of lines.

Example

```
%HI_RefreshData(_browser) ; updating (without changes in
condition, sorting or parameters)
%HI_RefreshData(_browser, "ID=21") ; changing the condition and
updating
%HI_RefreshData(_browser, "ID=21", "ID") ; changing the condition,
sorting and updating
```

```

INT _cnt ; the local variable for the parameterized condition
TEXT _name ; the local variable for the parameterized condition
TEXT _INVALID = 1/0 ; auxiliary invalid constant of text type

[cnt := 1]
[name := "%SELF%"]

; 1. change of condition (parameterized), sorting, parameters and updating
%HI_RefreshData(_browser, "CNT=#PAR# AND NAME LIKE #PAR#", "NAME", _cnt,
_name)

[cnt := 2]
[name := "%ODBC%"]

; 2. just the change of parameters (recycling the SQL expression) and
; updating
%HI_RefreshData(_browser, "CNT=#PAR# AND NAME LIKE #PAR#", "NAME", _cnt,
_name) ;
%HI_RefreshData(_browser, _INVALID, _INVALID, _cnt, _name) ; equivalent
alternative to the writing by line above (if the condition has not been
changed from the point 1)

; 3. without changes (recycling the SQL expression) and updating
%HI_RefreshData(_browser, "CNT=#PAR# AND NAME LIKE #PAR#", "NAME", _cnt,
_name)
%HI_RefreshData(_browser) ; equivalent alternative to the writing by line
above (it the condition, sorting and parameters have not been changed from
point 2)

```



Related pages:

- [Graphic object manipulation functions](#)
- [Function arguments - types](#)