


Text Entry Field


Windows control - Text entry field

Windows control - The text field is used to display and enter information in a text or numeric forms. It also allows us to work with the information of logical type, but Windows control of the [Check box type](#) is better for displaying logical information.

Example



To create a text entry field

1. In the [Drawing toolbar](#) click the button **Text entry field** .
2. Point the mouse cursor to the position of the first displayer corner and click.
3. Point the mouse cursor to the position of the diametric displayer corner and click.

Text entry field parameters are being configured in the palette [Connect object](#) through the following tabs:

Connect

Connected object

Selection of an object to connect. The object determines the contents of the text entry field and vice versa - the value of the object is changed when the user enters a new value into the text entry field and clicks another graphic object in the picture. This feature does not apply in the case of the parameter [RichEdit](#). The type of displayed value is given by the type of object.

When you connect an object to a *Structured variable*, you must define whose item will be displayed in the displayer. Unique item position in a structured variable is defined by the column name (the parameter **Column**) and the row number (the parameter **Row**). For an object of the *Value array*, you must define which array item will be displayed in the displayer.

Script

Reference variable

Specifies the name of the text entry field in the form of a reference variable for use in the picture script.

Metadata

Assignment of application data ([metadata](#)) to a graphic object. If the object contains some metadata, the name of the first record will be displayed on the button. Multiple records are indicated by ",...".

Event

The list box allows the user to select an [active picture event](#) for the text entry field. The list box contains the following active picture events:

- [OnClick](#)
- [OnGotFocus](#)
- [OnLostFocus](#)
- [OnMouseDown](#)
- [OnTextChange](#)
- [OnTypeTextChange](#)

Event Handler

The button opens the active picture script to define the active picture event specified by the parameter **Event**.

Info text/URL

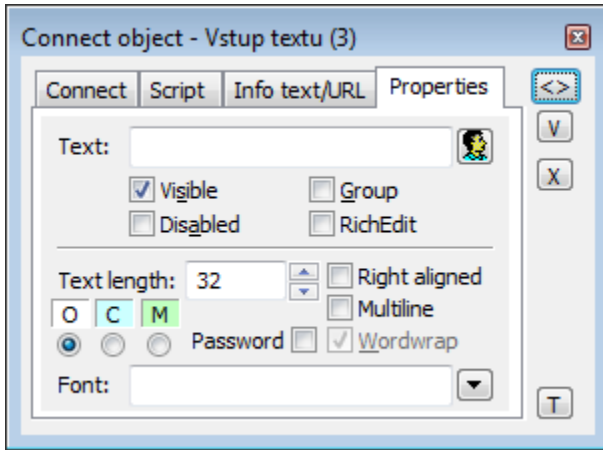
Info text

Text to display as a tooltip when the user points the mouse cursor to the text entry field in the picture opened in the [D2000 HI](#) process. Possibility to use the [Dictionary](#) (to open press **CTRL+L**).

URL




Definition of URL address to open a web page from a picture. The address may be set also in the picture script by the function [%HI_SetOBJURL](#).

Properties



The text entry field may be conditioned that it must be filled in HI. If any text is not entered, the background of the field matches the color defined in the application parameters. In the D2000 GrEditor, the entry field is always colored according to this parameter.

Conditions for using background:

-  Optional item - no change of color (default),
-  Conditioned mandatory item – color set according to application parameters [ColorConditionalItems](#)
-  Mandatory item – color set according to application parameters [ColorMandatoryItems](#)

Text

The parameter is useless for this type of Windows control.

Visible

If the parameter is checked, the text entry field will be visible after first opening the picture in the [D2000 HI](#) process. The parameter can be controlled from the script using the function [%HI_SetVisible](#).

Disabled

If the parameter is checked, the text entry field will be disabled after first opening the picture in the [D2000 HI](#) process. The parameter can be controlled from the picture script using the function [%HI_SetEnable](#).

Group

The parameter allows you to set the text entry field as the first control of a [group of Windows controls](#).

RichEdit

The parameter allows opening a window controller *RichEdit* instead of the *Text entry field*. It enables editing text and the use of more functions like more fonts and font colors. Text may be modified in the picture script by the [functions](#).

The shortcuts are supported directly when you are writing some text: **Ctrl+B** - bold, **Ctrl+U** - underline, **Ctrl+I** - italic.

RichEdit enables 2 text types: **formatted** (it contains the format commands which control the way to display a text) and **plain** text without formatting.

For that reason, it is valid:

If some object is connected to *EntryField*, its value is visible, but a change, made by the user, is not automatically set to this variable.

If some text **without formatting** is set into the variable, the formatting would disappear after leaving the Windows control - a new value of the variable would substitute it.

If the text **with formatting** is set, the text would be unreadable by other means. It depends on an application to read the content when the focus loses or doing other activities.

Text length

Definition of the maximum number of characters, which can be entered into the text entry field.

Password

Displays an asterisk (*) for each character typed into the text entry field.

Right aligned

If the parameter is checked, the text is right-aligned within the text entry field.

Multiline

The parameter allows to display and insert of multiline text.

Wordwrap

Automatically wraps a string typed into the text entry field after it reaches the right side of the field (the parameter **Multiline** must be also checked).

Font

Clicking the button placed right opens a list of text fonts. A font selected from the list will be used for the text entry field. For more information on text fonts see the topic [Text fonts](#).

Note: Font can be modified (changed) from the picture script using the function [%HI_SetFontStyle](#).

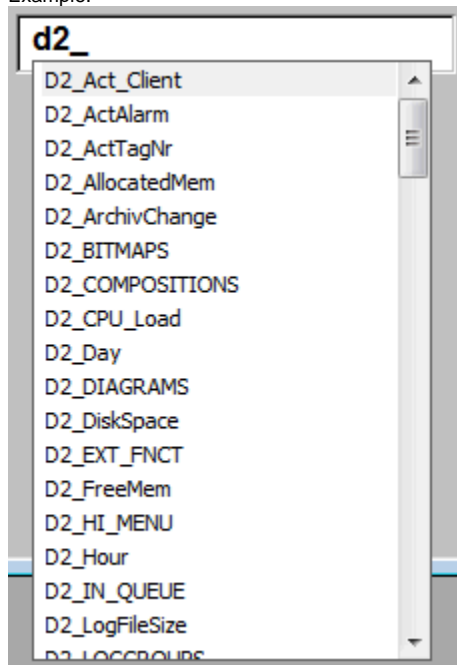
AutoComplete List

For the Entry field, there is allowed to assign AutoComplete List (text prediction, list of words). Users may choose a word from the list and insert it in the defined position in the *Entry field*. The items are defined on the application side, which enables their different utilization. The primary function of AutoComplete List is the possibility to add words and insert them into the written text. This feature is supported by ESL function [%HI_ShowAutoCompleteList](#). [OnTypeTextChange](#) picture event enables showing the written text. [OnAutoCompleteListSelected](#) picture event enables showing the selected text.

Features:

- Filling the list
- Selection of the item from the list and insertion to the *Entry field* (from the defined position).
- Handling:
 - creating the list - [%HI_ShowAutoCompleteList](#) function,
 - moving in the list with up/down arrows,
 - selection from the list - with Enter key or left mouse double click,
 - cancelling the list - Esc key, right and left arrows, mouse-clicking out of the list.
- The list appears from the given position `_startPos` which is defined by [%HI_ShowAutoCompleteList](#).

Example:



```

ENTRYX OnTypeTextChange(IN INT _refID, IN TEXT _Line, IN INT _StartPos, _CursorPos)

INT _i
INT _count
INT _iDim

TEXT _tPrefix
TEXT _tDBwhere

RECORD (SD.ARR_TEXT) _tRec
RECORD (SD.DICTIONARY) _tRecDB

IF 0 < %LenStr(_Line) THEN
  _tPrefix := %SubStr(_Line,_StartPos,_CursorPos - _StartPos)
  IF 0 < %LenStr(_tPrefix) THEN
    _tDBwhere := "NAME LIKE '" + _tPrefix + "%'"
    RECORD DB.DICTIONARY, _tRecDB, _retCode, 1000 WHERE _tDBwhere

    IF _retCode = 0 THEN
      _iDim := _tRecDB\DIM

      REDIM _tRec[_iDim]

      COPYCOL _tRecDB^NAME, _tRec^TEXT1

      SORT _tRec^TEXT1, @TRUE

      %HI_ShowAutoCompletionList(_refID,_tRec^TEXT1, _StartPos)

    ENDIF
  ENDIF
ENDIF

END OnTypeTextChange

```



Related pages:

[Windows controls](#)

[Graphic object manipulation functions](#)