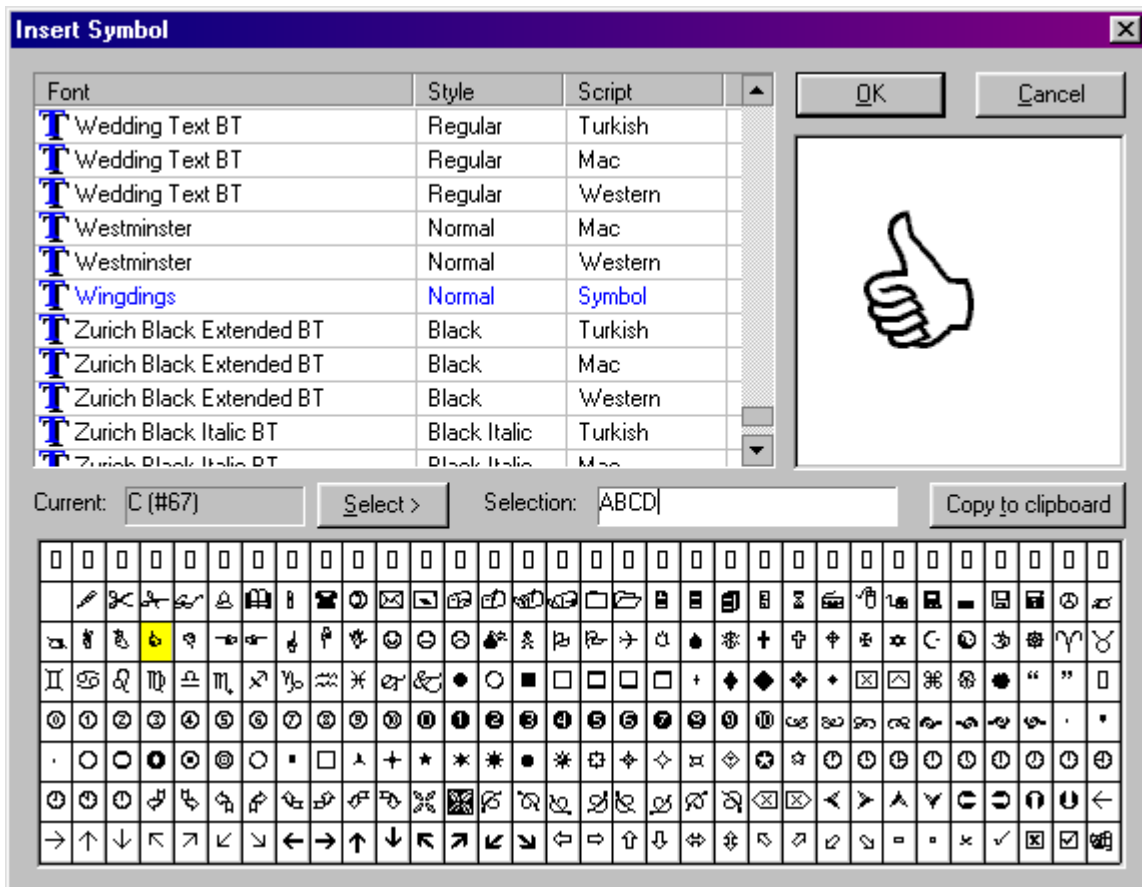




OCX Controls & software technologies
<http://www.precisionocx.com>

Precision Symbol Picker

Version 1.1



Overview

Purpose

Precision Symbol Picker is a powerful OCX control for your applications which displays the listing of all the fonts installed on the system, the complete character map for each of them and lets the user choose a specific character. It provides the sum of the features you can find in the "Insert Symbol" dialog box present in many word processing programs (e.g. *Microsoft Word*) and in the "Character map" program shipped with *Windows*.

Features

Precision Symbol Picker displays a dialog window that is larger and more detailed than the "Character map" or the "Insert Symbol" window. A full-featured list control displays all the fonts installed in the system, an icon tells the type of each font: TrueType, raster or device.

A large, high resolution area is reserved for the zoomed version of the currently selected character. The user can easily choose the desired font in the list control by using the mouse or by using the up, down, page up, page down arrow keys. At the same time, with the right and left key it is possible to switch to the following or preceding symbol.

The user can simply choose one (or more character) for use in your program, or he can copy them into the clipboard.

High customisability

You can customize the title of the window, the size of the characters displayed both in the character map and in the zoom area, the name of the font that is initially chosen when the window appears, and the character that is initially highlighted.

The possibility to choose more than one character can be disabled and it is possible to choose whether to show the "Copy to clipboard" button or not.

Optimization

Precision Symbol Picker was written entirely in C++ and is composed of strongly optimized code. The only required libraries are the MFC and MSVCRT dll that are shipped with every version of Microsoft Windows. You do not need any additional libraries.

Small size

Precision Symbol Picker is 60k in size ! This means that it is very fastly loaded into your application and it does not significantly increase your final application size.

Any special requirement ?

If you are developing an application and found Precision Symbol Picker nearly perfect for you needs except for small details, please feel free to contact us: let's talk about it !

Documentation

All the documentation you may need on Precision Symbol Picker is on line !

Quick start

To try the look and feel of the Symbol Picker dialog window, run the provided **TrySymbolPicker** test program.

To start using the component in your programs:

- | if you purchased a license, call the *SetLicense(...)* using your license number as argument;
 - | call the *InvokePickerDialog()* method and store the boolean result
 - | if the result is TRUE, the user has pressed the **OK** button
 - | the selected symbol (or symbols) are in the property string *ChosenChar*
 - | the selected font face is in the property string *ChosenFontFace*
-

Evaluation use

Symbol Picker is provided in two different versions: a trial version and a full version.

The trial version is freely distributable and has the following restrictions:

- | informational dialog windows are displayed when the object starts running;
- | the "Ok" and "Copy to clipboard" won't store or copy the characters you have actually chosen but a fixed string telling that the control is in trial version.
- | the dialog window will have an "***Evaluation version***" caption.

The full version is reserved for customers who bought the product and own a license number. The control works exactly like the trial version as far as the serial number is specified in a call to the *SetLicense()* function is issued. After that all the functions of the control are unlocked and work as documented below.

Properties

AllowMultipleChars

Allowed values: true, false

Default value: true

Meaning: this property tells whether the user will be allowed to select more than one character in the window. If this property is set to **true**, when the user press the "Select" button, the currently highlighted character will be appended to the current selection string; if it is set to **false**, when the user press the "Select" button, the currently highlighted character will replace the current selection.

ChosenChar

Allowed values: any string

Meaning: after the window has been closed, this string is filled with the character (or the characters, if *AllowMultipleChars* is enabled) chosen by the user. If the user did not choose any character, this string is empty.

ChosenFontFace

Allowed values: any string representing a font name

Meaning: this property contains the name of the font chosen by the user. You should access this property after the control's dialog window is closed by the user.

EnableToClipboard

Allowed values: true, false

Default value: true

Meaning: this property tells whether the "Copy to clipboard" button has to be shown on the window or not.

InitialChar

Allowed values: any string

Default value: 'A'.

Meaning: this is the character initially highlighted when the window appears. If you provide a string of more than one character, only the first character will be used.

InitialFontFace

Allowed values: any string representing a font name

Default value: Wingdings

Meaning: this is the name of the font initially set when the window appears. It is not required to provide the full name, you can specify an initial substring only. Make sure that the name you provide actually leads to a font that is really present on the system, otherwise the control will make the choice for you.

LargeFontSize

Allowed values: any integer number

Default value: 120

Meaning: this is the size of the font used in the zoom area. If you increase this value too much the character could be graphically truncated.

SmallFontSize

Allowed values: any integer number

Default value: 14

Meaning: this is the size of the font used in the character map area.

WindowCaption

Allowed values: any string

Default value: "Insert Symbol"

Meaning: this is the caption of the dialog window.

Methods

boolean InvokePickerDialog()

Parameters: none

Return value: *true* if the user chose "OK", *false* if he chose "Cancel"

Meaning: call this method to actually show the window and process user choices. Remember to set the desired properties before calling this method.

boolean SetLicense(long LicenseNumber)

Parameters: LicenseNumber is the number you received

Return value: *true* if the unlocking is successful, *false* otherwise.

Meaning: call this method as the first thing, if you purchased a license, in order to unlock all of the component's features. **This method only works with the full version:** it has no effect if used on the trial version.

Version History

Version 1.1

Released the two different versions "trial" and "full" with minor differences. Additionally, a bug was removed which prevented the full release of all the allocated GDI resources. The bug could have lead, in the long run, to a shortage in the availability of the system's GDI resources.

Version 1.0

Initial release.

Precision Symbol Picker vs. Character Map

