

About Polar AutoCorrect

Polar AutoCorrect consists of two components: ActiveX control (AutoCorrect.ocx) and Dynamic-link library (AutoCorrectDLL.dll). It allows you to add AutoCorrect capabilities to your application.

Polar AutoCorrect is primarily a tool for adding AutoCorrect functionality to a higher-level application. Using it is a matter of passing in word for AutoCorrect and reading the corrected word. Standard features include:

- Correcting two initial caps.
- Capitalizing of the first letter in the sentence.
- User defined exceptions for capitalizing of the first letter in the sentence.
- Correcting of accidental use of CAPS LOCK key.
- User definable words for replacing while typing.
- Highly customizable
- Dialogs can be easily translated to any language

This Help introduces all functions (properties and methods) implemented in Polar AutoCorrect ActiveX control. Functions implemented in AutoCorrect DLL have the same description and its syntax is described in

autocorr.h	header file for C++ developers
AutoCorrect.txt	text file for API Text Viewer (for Visual Basic developers)

Technical Support

If you need help using **Polar AutoCorrect**, contact Polar in any of the following ways:

- visit our home page at www.polarsoftware.com.
- E-mail: support@polarsoftware.com or info@polarsoftware.com.

Using Polar AutoCorrect with Visual Basic

Adding POLAR AutoCorrect control to Toolbox:

From "Project" menu select "Components..." or press Ctrl + T

In "Components" dialog find "POLAR AutoCorrect ActiveX Control module" in the list

Select it and click checkbox next to it

Click "Apply" and then "OK" to close dialog

Now there will be new icon for POLAR AutoCorrect in the toolbox

Using POLAR AutoCorrect control

1. Create new standard project
2. Insert POLAR AutoCorrect component into the form
3. Create five buttons and change their caption to "Initialize", "About", "DlgBox", "GetCorrectWord", "CapitalizeException"
4. Create one text box
5. For button "Initialize" put this code:

```
Private Sub Command1_Click()  
    Dim ini As String  
    Dim def As String  
    def = "c:\test.acr"  
    ini = "c:\test.ini"  
    AutoCorrect1.Initialize def, ini  
End Sub
```

6. For button "About" put this code:

```
Private Sub Command2_Click()  
    AutoCorrect1>AboutBox  
End Sub
```

7. For button "DlgBox" put this code:

```
Private Sub Command3_Click()  
    AutoCorrect1.DlgBox  
End Sub
```

8. For button "GetCorrectWord" put this code:

```
Private Sub Command4_Click()  
    Dim S As String  
    AutoCorrect1.GetCorrectWord Text1.Text, S  
    Text1.Text = S  
End Sub
```

9. For button "CapitalizeException" put this code:

```
Private Sub Command5_Click()  
    If AutoCorrect1.IsCapitalizeException(Text1.Text) Then  
        Text1.Text = "True"  
    Else  
        Text1.Text = "False"  
    End If  
End Sub
```

10. Run application

11. Click "Initialize"

12. Click "About"

13. Click "DlgBox"

In "Replace" field write "polar"

In "With" field write "Polar"

Click "Add"

Click "Exceptions..."

In textbox write "etc." and click "Add"

Close both dialogs

14. Write "polar" in textbox

15. Click "GetCorrectWord"

Word "Polar" will appear in the textbox

16. Write "etc." in textbox

17. Click "CapitalizeException"

"True" will appear in the textbox

18. Write "HOuse" in textbox

19. Click "GetCorrectWord"

Word "House" will appear in the textbox

20. Turn on "Caps Lock"

21. Write "hOUSE" in textbox

22. Click "GetCorrectWord"

Word "House" will appear in the textbox

Using Polar AutoCorrect with Microsoft Visual C

Adding POLAR AutoCorrect control:

From "Project" menu select "Add To Project" / "Components and Controls"

In "Registered ActiveX Controls" directory select "POLAR AutoCorrect Control"

Click button "Insert"

Click "OK" on next dialogs

Now there will be new icon for POLAR AutoCorrect in the "Controls" palette

Using POLAR AutoCorrect control

1. Create new MFC AppWizard project (Dialog Based)
2. Insert POLAR AutoCorrect component into the form
3. Create five buttons and change their caption to "Initialize", "About", "DlgBox", "GetCorrectWord", "CapitalizeException"
4. Create one edit box
5. Add member variable (Category: control) for AutoCorrect control and name it m_AutoCorrect
6. Add member variable (Category: Value, Variable Type: CString) for edit box and name it m_edit
7. For button "Initialize" put this code:

```
VARIANT str ;  
str.vt = VT_BSTR;  
str.bstrVal = (unsigned short*) "c:\\test.ini";  
m_AutoCorrect.Initialize( "c:\\test.acr", str );
```

8. For button "About" put this code:

```
m_AutoCorrect.AboutBox();
```

9. For button "DlgBox" put this code:

```
m_AutoCorrect.DlgBox();
```

10. For button "GetCorrectWord" put this code:

```
UpdateData(TRUE);  
BSTR str ;  
m_AutoCorrect.GetCorrectWord( m_edit, &str );  
m_edit = str;  
UpdateData( FALSE );
```

11. For button "CapitalizeException" put this code:

```
UpdateData( TRUE );  
if ( m_AutoCorrect.IsCapitalizeException( m_edit ) )
```

```
m_edit = "True";  
else  
m_edit = "False";  
UpdateData( FALSE );
```

12. Run application

13. Click "Initialize"

14. Click "About"

15. Click "DlgBox"

In "Replace" field write "polar"

In "With" field write "Polar"

Click "Add"

Click "Exceptions..."

In edit box write "etc." and click "Add"

Close both dialogs

16. Write "polar" in textbox

17. Click "GetCorrectWord"

Word "Polar" will appear in the edit box

18. Write "etc." in textbox

19. Click "CapitalizeException"

"True" will appear in the edit box

20. Write "HOuse" in textbox

21. Click "GetCorrectWord"

Word "House" will appear in the edit box

22. Turn on "Caps Lock"

23. Write "hOUSE" in edit box

24. Click "GetCorrectWord"

Word "House" will appear in the edit box

Using Polar AutoCorrect with Borland C++ Builder 1.0

Adding POLAR AutoCorrect control to tool palette:

From "Component" menu select "Install..."

In "Install Components" dialog click "ActiveX" button

Find "POLAR AutoCorrect ActiveX Control module" in the list and select it

Click "OK" on both dialogs

Now there will be new icon for POLAR AutoCorrect in the palette (Active X)

Using POLAR AutoCorrect control

1. Create new project
2. Insert POLAR AutoCorrect component into the form
3. Create five buttons and change their caption to "Initialize", "About", "DlgBox", "GetCorrectWord", "CapitalizeException"
4. Create one edit box

5. For button "Initialize" put this code:

```
AutoCorrect1->Initialize("c:\\test.acr", "c:\\test.ini");
```

6. For button "About" put this code:

```
AutoCorrect1->AboutBox();
```

7. For button "DlgBox" put this code:

```
AutoCorrect1->DlgBox();
```

8. For button "GetCorrectWord" put this code:

```
AnsiString str1, str2;  
str1 = Edit1->Text;  
AutoCorrect1->GetCorrectWord(str1 , str2);  
Edit1->Text = str2;
```

9. For button "CapitalizeException" put this code:

```
if (AutoCorrect1->IsCapitalizeException(Edit1->Text))  
    Edit1->Text = "TRUE";  
else  
    Edit1->Text = "FALSE";
```

10. Run application

11. Click "Initialize"

12. Click "About"

13. Click "DlgBox"

In "Replace" field write "polar"

In "With" field write "Polar"

Click "Add"

Click "Exceptions..."

In edit box write "etc." and click "Add"

Close both dialogs

14. Write "polar" in edit box

15. Click "GetCorrectWord"

Word "Polar" will appear in the edit box

16. Write "etc." in edit box

17. Click "CapitalizeException"

"True" will appear in the edit box

18. Write "HOuse" in edit box

19. Click "GetCorrectWord"

Word "House" will appear in the edit box

20. Turn on "Caps Lock"

21. Write "hOUSE" in edit box

22. Click "GetCorrectWord"

Word "House" will appear in the edit box

Using Polar AutoCorrect with Borland Delphi 3.0

Adding POLAR AutoCorrect control to tool palette:

From "Component" menu select "Import ActiveX Control..."

Find "POLAR AutoCorrect ActiveX Control module" in the list and select it

Click "Install"

Click "OK" in "Install" dialog

Click "OK" for rebuilding

Close "Package" dialog and save changes

Now there will be new icon for POLAR AutoCorrect in the palette (Active X)

Using POLAR AutoCorrect control

1. Create new project
2. Insert POLAR AutoCorrect component into the form
3. Create five buttons and change their caption to "Initialize", "About", "DlgBox", "GetCorrectWord", "CapitalizeException"
4. Create one edit box

5. For button "Initialize" put this code:

```
var acr, ini:OleVariant;  
begin  
  AutoCorrect1.Delphi := TRUE;  
  acr := 'c:\default.acr';  
  ini := 'adef.ini';  
  AutoCorrect1.Initialize(acr, ini);  
end;
```

6. For button "About" put this code:

```
AutoCorrect1.AboutBox;
```

7. For button "DlgBox" put this code:

```
AutoCorrect1.DlgBox;
```

8. For button "GetCorrectWord" put this code:

```
var str2:WideString;  
begin  
  AutoCorrect1.GetCorrectWord(Edit1.Text, str2);  
  Edit1.Text := str2;  
end;
```

9. For button "CapitalizeException" put this code:

```
if (AutoCorrect1.IsCapitalizeException(Edit1.Text)) then  
    Edit1.Text := 'TRUE'  
else  
    Edit1.Text := 'FALSE';
```

10. Run application

11. Click "Initialize"

12. Click "About"

13. Click "DlgBox"

In "Replace" field write "polar"

In "With" field write "Polar"

Click "Add"

Click "Exceptions..."

In edit box write "etc." and click "Add"

Close both dialogs

14. Write "polar" in edit box

15. Click "GetCorrectWord"

Word "Polar" will appear in the edit box

16. Write "etc." in edit box

17. Click "CapitalizeException"

"True" will appear in the edit box

18. Write "HOUse" in edit box

19. Click "GetCorrectWord"

Word "House" will appear in the edit box

20. Turn on "Caps Lock"

21. Write "hOUSE" in edit box

22. Click "GetCorrectWord"

Word "House" will appear in the edit box

void Initialize(LPCTSTR pszACorrectFileName, const VARIANT & pszINIFilename)

Visual Basic:	Initialize String, String
Borland Delphi:	Initialize(WideString, OleVariant)
Borland C++ Builder:	Initialize(AnsiString, AnsiString)

pszACorrectFileName	pointer to a string that holds name of AutoCorrect settings file (full path)
pszINIFilename	VARIANT that holds name of AutoCorrect ini file (full path)

Remarks

Initialize must be called first.

AutoCorrect settings file is used for storing words and their replacements, and capitalizing exceptions.

AutoCorrect ini file is used for storing states of options in DlgBox (CapitalizeSentence, CorrectCapsLock etc.)

If any of these files does not exist, it will be created.

BOOL GetCorrectWord(LPCTSTR pszTestString, BSTR *pbstrCorrectWord)

Visual Basic:	GetCorrectWord String, String
Borland Delphi:	GetCorrectWord(WideString, WideString)
Borland C++ Builder:	GetCorrectWord(AnsiString, AnsiString)

pszTestString pointer to a string that holds word for testing

pbstrCorrectWord pointer to a string that will receive corrected word

Return value

TRUE if word is corrected

FALSE if word is not corrected

Remarks

Use this method for correcting words. If word needs to be corrected, correction will be passed in second string. Otherwise second string will be empty.

List of words and their corrections can be modified in "Options Dialog box".

See Also:

[DlgBox](#)

[IsCapitalizeException](#)

BOOL IsCapitalizeException(LPCTSTR pszTestWord)

Visual Basic: IsCapitalizeException String

Borland Delphi: IsCapitalizeException(WideString)

Borland C++ Builder: IsCapitalizeException(AnsiString)

pszTestWord pointer to a string that holds word for testing

Return value

TRUE if word is in capitalize exceptions list

FALSE if word is not in capitalize exceptions list

Remarks

Use this method for checking should next word be capitalized. If return value is TRUE than test word is in exception list and next word should not be capitalized.

Exception list can be modified in "Options Dialog box", under "Exceptions".

See Also:

[DlgBox](#)

[GetCorrectWord](#)

voidDlgBox()

Visual Basic:DlgBox

Borland Delphi:DlgBox

Borland C++ Builder:DlgBox()

Remarks

Shows AutoCorrect options dialog box.

void AboutBox()

Visual Basic: AboutBox

Borland Delphi: AboutBox

Borland C++ Builder: AboutBox()

Remarks

Shows Polar AutoCorrect about box.

void SetCapitalizeSentence(BOOL bNewValue)

BOOL GetCapitalizeSentence()

Visual Basic: CapitalizeSentence

Borland Delphi: CapitalizeSentence

Borland C++ Builder: CapitalizeSentence

bNewValue new state for "Capitalize Sentence" property

Return value

state of "Capitalize Sentence" property

Remarks

Use this property for checking or setting if capitalizing exceptions should be checked. If return value is TRUE then exception testing should be done.

See Also:

[CorrectCapsLock](#)

[CorrectTwoInitCaps](#)

[ReplaceText](#)

void SetCorrectCapsLock(BOOL bNewValue)

BOOL GetCorrectCapsLock()

Visual Basic: CorrectCapsLock

Borland Delphi: CorrectCapsLock

Borland C++ Builder: CorrectCapsLock

bNewValue new state for "Correct Caps Lock" property

Return value

state of "CorrectCapsLock" property

Remarks

Use this property for checking or setting if accidental usage of CAPS LOCK will be corrected. If return value is TRUE than correcting will be done.

See Also:

[CapitalizeSentence](#)

[CorrectTwoInitCaps](#)

[ReplaceText](#)

void SetCorrectTwoInitCaps(BOOL bNewValue)

BOOL GetCorrectTwoInitCaps()

Visual Basic: CorrectTwoInitCaps

Borland Delphi: CorrectTwoInitCaps

Borland C++ Builder: CorrectTwoInitCaps

bNewValue new state for "Correct Two Init Caps" property

Return value

state of "Correct Two Init Caps" property

Remarks

Use this property for checking or setting if accidental two init caps will be corrected. If return value is TRUE then correcting will be done.

See Also:

[CapitalizeSentence](#)

[CorrectCapsLock](#)

[ReplaceText](#)

void SetReplaceText(BOOL bNewValue)

BOOL GetReplaceText()

Visual Basic: ReplaceText

Borland Delphi: ReplaceText

Borland C++ Builder: ReplaceText

bNewValue new state for "Replace Text" property

Return value

state of "Replace Text" property

Remarks

Use this property for checking or setting if correcting of misspelled words will be used. If return value is TRUE than correcting will be done.

See Also:

[CapitalizeSentence](#)

[CorrectCapsLock](#)

[CorrectTwoInitCaps](#)

void SetDelphi(BOOL bNewValue)

BOOL GetDelphi()

Borland Delphi: Delphi

bNewValue new state for "Delphi" property

Remarks

Use this property for checking or setting if working environment is Borland Delphi.

In Delphi add line to your code to set this property to TRUE:

```
SpellChecker1.Delphi := TRUE;
```

before using any other method. In other developing environments this property does not need to be changed.

AutoCorrect Options Dialog properties

StrACDialogCaption

StrACDlgCorrectTwoCaps

StrACDlgCapitalizeFirstLetter

StrACDlgCorrectCapsLock

StrACDlgReplaceText

StrACDlgReplace

StrACDlgWith

StrACDlgButtonClose

StrACDlgButtonExceptions

StrACDlgButtonAdd

StrACDlgButtonChange

StrACDlgButtonDelete

By default all text in AutoCorrect dialog is in English. With these properties dialog can be translated into any language. If value of property is an empty string, than default (English) will be used, otherwise value of property will be used for text in dialog.

See Also:

[Exceptions Dialog properties](#)

Exceptions Dialog properties

StrExcDlgCaption

StrExcDlgDontCapitalize

StrExcDlgButtonClose

StrExcDlgButtonAdd

StrExcDlgButtonDelete

By default all text in Exceptions dialog is in English. With these properties dialog can be translated into any language. If value of property is an empty string, than default (English) will be used, otherwise value of property will be used for text in dialog.

See Also:

[AutoCorrect Options Dialog properties](#)

