Subject: RFC : my first ESC macros
Posted by gprentice on Sat, 10 Dec 2005 12:02:17 GMT
View Forum Message <> Reply to Message

I've written my first ESC macros for U++ !

ESC and the editor macro mechanism are very nice!!

After establishing that the IDE editor didn't have Find next word bound to Ctrl+F3 (the key I usually use for this) (ahaha), I went ahead and wrote my own. Then when I tried to bind it to Ctrl F3 I found that theIDE already had it! Darn! Oh well, my one works better anyway coz it wraps in the file

I would like to be able to use global variables to get persistent values but I don't know how. Also CopyWord and InsertLine don't work because they need clipboard access.
Also I'm wondering if I can have global is_wordchar().
Also a way to list existing key bindings would be nice!

Graeme

```
macro "Swap":"Swap Chars" Ctrl+Shift+T {
 pos1 = GetCursor();
 if (GetColumn(pos1)==0)
 {
    pos1 += 1;
        SetCursor(pos1);
 }
 s = Get(pos1);
 Remove(1);
 SetCursor(pos1-1);
 Insert(s);
 SetCursor(pos1);
}

// need some clipboard access for this to work
macro "Copy":"Copy Word" Ctrl+K {
 #:is_wordchar(ch) {
  return (ch >= 'A' && ch <= 'Z') ||
        (ch >= 'a' && ch <= 'z') ||
        (ch >= '0' && ch <= '9') || (ch == '_');
 }
 pos1 = GetCursor();
 s = Get(pos1);
 if (!is_wordchar(s[0]))
  return;
```

```
 MoveWordRight();
 MoveWordLeft();
   MoveWordRight(1); // select word
   // CopySelection();  // how do I do this? I know, I'll modify upp source!
}

// Intelligent line insert - moves cursor to col zero before pasting
// but needs clipboard access
macro "Insert":"Insert Line" Ctrl+Shift+L {
 pos1 = GetCursor();
 SetCursor(pos1 - GetColumn(pos1));
 // PasteClipboard();
}


// how do I use global variables to get persistent values?
macro "Find":"Find Next Word" Ctrl+Shift+W {
 #:is_wordchar(ch) {
  return (ch >= 'A' && ch <= 'Z') ||
         (ch >= 'a' && ch <= 'z') ||
         (ch >= '0' && ch <= '9') || (ch == '_');
 }
 pos1 = GetCursor();
 s = Get(pos1);
 MoveWordRight();
 if (!is_wordchar(s[0]))
  return;

 pos2 = GetCursor();
 MoveWordLeft();
 pos3 = GetCursor();
 s2 = Get(pos3, pos2 - pos3);
 MoveWordRight();
 if (Find(s2,1,0,0))
  MoveWordLeft();
 else {
    // wrap to top of file
    // BeepOnWrap(); !
  MoveTextBegin();
  if (Find(s2,1,0,0))
   MoveWordLeft();
  else {
     SetCursor(pos1); // unexpected!
     return;
  }
 }
 // would like to select the whole string here but it means
```

```
 // cursor moves off the word so next find doesn't work
 // - anyway, selection of one character seems ok
 pos4 = GetCursor();
 SetSelection(pos4, 1);
 // next time in I can use GetSelCount to tell whether a previous
 // find next word was done and if so, to search for the string already in s2
 // so that it doesn't extend the length of the word and fail to get back
 // to the original starting point which was a shorter word
}
```

Subject: Re: RFC : my first ESC macros
Posted by mirek on Sat, 10 Dec 2005 13:56:25 GMT
View Forum Message <> Reply to Message

gprentice wrote on Sat, 10 December 2005 07:02
I've written my first ESC macros for U++ !

After establishing that the IDE editor didn't have Find next word bound to Ctrl+F3 (the key I usually use for this) (ahaha), I

macro "Swap":"Swap Chars" Ctrl+Shift+T {

Actually, there is unresolved issue with macro keys... The problem is that they are hardwired in the macro definition - that is good as long as you do not want to distribute macros, but cames short if we would like to add some macros to existing packages... (because users are free to redefine some key to the key assigned to macro).

I am not sure what is the right solution here....

Subject: Re: RFC : my first ESC macros
Posted by gprentice on Sat, 10 Dec 2005 22:04:18 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 10 December 2005 08:56

Actually, there is unresolved issue with macro keys... The problem is that they are hardwired in the macro definition - that is good as long as you do not want to distribute macros, but cames short if we would like to add some macros to existing packages... (because users are free to redefine some key to the key assigned to macro).

I am not sure what is the right solution here....

I'm not sure I understand. Are you saying that it should be possible to assign a key to a macro without having to change the macro source?

If so, I'm wondering what kind of macros these would be e.g. editor functions that are worth distributing as macros would more likely be provided as built in (C++ code). If somebody takes the FindNextWord macro I wrote and changes it to assign their own key, then I put out another version, they have to reassign the key again in the macro code. I guess it would be nicer if there was a dialog (or even just a text file) that listed macro name : key assignment that overrode the macro code - it doesn't seem like a big problem though. Am I on the right track of what you're saying here?

I don't have a clear picture of how key bindings are handled at present and what overrides what. At some point, I'll dig into the source.

Did you notice my question about global variables in editor macros - is that possible at present - (it's ok if it's not)

Graeme

Subject: Re: RFC : my first ESC macros
Posted by mirek on Sat, 10 Dec 2005 22:28:17 GMT
View Forum Message <> Reply to Message

Well, as macros can be bound to packages, it could have sense to provide macros to e.g. speedup creation of dialogs (and put them to CtrlLib) etc... OTOH, you are right that the same thing can be as effectively done in TheIDE C++ code - but once again, that is less possible for third party package (if there will be any ever

As for global variables, I am afraid that they are not available between two macro runs. However, Esc provides global variables - see Esc reference- but they will exist just for single run.
