UniView plugin SDK reference manual

Revision 1.0, copyright ©Andrej Krutak 2001
questions: krutak@gym.utcru.sk
homesite: http://www.gym.utcru.sk/andrek/

What are plugins

Plugins are small (or big, it depends on authors :) programs, which are extending functionality of some other program (in this case UniView). In following document, we will talk just about UniView plugins.
UniView plugins are standard windows DLL's renamed to UVP files. They MUST be placed in '*plugins*" directory and export one special function in order to be recognized as a UniView plugin...

How to create plugin

You can create plugin in two ways:
1st: You can use generic plugin workspace
2st: You can create DLL file with special exported function

If you want to use first step, you must have Visual C++ 6.0 installed and all should work just fine. If you have VC++ 5.0 installed, it should also work, but you must create workspace by yourself...
If you don't have VC++ or want to use 2nd step, you must create a DLL with following attributes:
» it must be a 32-bit DLL
» it must export function with name exactly '*_uvplugin_main*'
» function uvplugin_main must meet all specifications from this manual

Plugin main function

Main function of each plugin is function uvplugin_main. It must have following format:

```
int __stdcall uvplugin_main(HINSTANCE, DWORD, DWORD, DWORD);
```

As you can see, it is a stdcall function. If it'll be eg. fastcall, it may cause crash of UniView, so please remember to always set this type of function!
» First parameter - HINSTANCE - always contains instance of plugin module.
» 2nd parameter contains message ID.
» 3rd and 4th parameter are containing parameters for messages.
All message IDs and related parameters are described later.

Function must return integer value. If it's successful, it should return 0, otherwise nonzero value...

Note that int is a four byte signed value and DWORD is a unsigned long value. HINSTANCE is a module handle declared in standard windows headers.

Communication between plugin and UniView

A declared set of messages can be used for plugin to communicate with UniView and vice versa. They are defined in **plugin.h** header, which's included in this package. If UniView is sending message to plugin, it uses uvplugin_main function. If you want to send message to UniView, you should use exported function

```
DWORD __stdcall UniView_doservice(DWORD, DWORD, DWORD);
```

This function is exported from UniView as _UniView_doservice.
» 1st parameter is message
» 2nd and 3rd parameter are parameters of message
Function returns DWORD value, you already should know, what it is :)

If you want to create plugin, uvplugin_main has to process at least most important

messages.

Messages from UniView to plugins

UniView can send many messages to plugin to inform him about his actions but also to retrieve some informations from plugin...

UVMSG_INITPLUGIN
parameters: up_initdata, -
return value: -
This message is sent to plugin when UniView wants to retrieve some informations about plugin. In the *wParam* parameter is pointer to *up_initdata* structure, which plugin must fill in as follows:

Item	*meaning*
plugin_name	Name of plugin. Should be in format *plugin name (lang1 | lang2 | lang3)*, where langn are shortcuts of supported languages. This format isn't required but it's very good for user :-)
items_count	Count of menu items, which plugin occupies
version_1	Major version of UniView (for current releases should be 1). Please keep this number up-to-date with current version of UniView.
version_2	Minor version of UniView (for current releases should be 73 or less). Please keep this number up-to-date with current version of UniView.
up_menuitems[n]	You can allocate up to 64 menu items in UniView's menu by filling-in this structure. *Name* member must contain menu item name and *place* it's placing in menu (one of *MENU_* constants defined in plugin.h file)

UVMSG_DOCUMENT_NEWOPENED
parameters: -
return value: -
Informs plugin, that new file was opened in UniView (i.e. there's a new current document).

UVMSG_DOCUMENT_SAVED
parameters: -
return value: -
Informs plugin, that current document was saved to disk.

UVMSG_SETTINGS_CHANGED
parameters: -
return value: -
Informs plugin, that UniView settings were changed. This means e.g. language change, other colors or whatever.

UVMSG_STOPEXIT
parameters: -
return value: -
Asks plugin, it UniView can quit. It yes, it should return 0 or nonzero value otherwise.

UVMSG_EXIT
parameters: -
return value: -
Asks plugin to terminate. If this message is recieved by plugin, it must terminate immediately!

UVMSG_COMMAND
parameters: int, -
return value: -
If some of plugins menu items was activated (i.e. pressed by user). Parameter *wParam* contains

a *int* number of command, as it was defined by *UVMSG_INITPLUGIN* message (index in *up_menuitems*). Plugin should show some dialog box or perform some action. It should work just like some built-in function of UniView :-)

Messages from plugins to UniView

If plugin wants to know something, it can send a message to UniView using `UniView_doservice` function. Allowed messages are defined in plugin.h:

PLGMSG_GETLANG
parameters: -
return value: int
Returns number of actual language used by UniView. Codes of languages are listed in plugin.h file.

PLGMSG_GETUVINST
parameters: -
return value: HINSTANCE
Returns instance of main UniView program.

PLGMSG_GETUVHWND
parameters: -
return value: HWND
Returns handle to UniView window.

PLGMSG_GETEWHWND
parameters: -
return value: int
Returns handle to editing window of UniView. Type of this window depends on current document type and thus it doesn't use the same messages for all types of documents. You may use this handle just after checking of current file type (can be retrieved using *PLGMSG_GETDOCTYPE*).

PLGMSG_GETDOCPATH
parameters: -
*return value: char **
Returns path of current file. You MUST NOT change variable returned.

PLGMSG_GETIMGCLASS
parameters: -
*return value: Image_*
Returns Image pointer.

PLGMSG_DOCUMENTCHANGED
parameters: -
return value: -
Notifies UniView that current document was changed. This message must be used when you do something with image, text of binary files, you know...

PLGMSG_IMAGEMAKEUNDO
parameters: -
return value: -
Notifies UniView that it should create a undo image from current image because current image is about to be changen.

PLGMSG_GETDOCTYPE
parameters: -
return value: int
Returns number specifying current file type. You have to use this value rather than determine file type by it's extension. If your plugin is working with specific file tye, you should test current file type before doing anything...

Notes

Please remember to put *plugin.h* and *cimage.h* files in path of your compiler if you want to use

them. You also have to link your files with uniview.lib file (for MS Visual C++ compiler). If you don't have VisualC++, I don't know how to produce plugins. If you have an idea, don't hositate and throw a email :-)

If you want your plugin to be translated to as many languages as possible, send me a mail. I can ask translators to translate your plugin :-).

Finish

If you want to see some template plugin, check directory "*generic*". If you have any questions about plugins, send me email :-D