
Subject: Re: ENVI menus, programmatically.
Posted by [Peter Mason](#) on Tue, 16 Mar 2004 22:14:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Robert Moss wrote:

> Does anyone here know the trick to programmatically adding menu items
> to ENVI? Yes, I have "googled" on this topic, and found one instance
> that looked promising, but it doesn't really answer the question.
> Clearly there is a way to do it... and it appears to involve creating
> a procedure that has one parameter which is a data structure that
> defines the menu structure. What happens inside this routine, or what
> undocumented programs it may call, or what undocumented program call
> it, is non-obvious. Any takers?
>
> Robert Moss, PhD
> EOIR Technologies Inc.
> robert.moss@nvl.DONOTSPAMTHEarmy.mil

I wish that RSI would get on with it and document this interface as it is very useful. IMHO, they under-estimate the nuisance factor involved in the traditional route of editing the menu file. Even if you give copy-and-paste instructions to your users, it's still a nuisance as they have to find the envi.men file, overcome the geek factor, and so on. Most ENVI users can't be having with this sort of run-around. (Other sorts, sure, but not this one :-))

Whereas.

I sort-of found out how this interface works a good few years ago and use it routinely. *It has been stable for a number of ENVI versions.* In the interests of motivating some genuine documentation, here's what I know:

The system only kicks in if your file (.PRO or .SAV) is in ENVI's SAVE_ADD subdirectory.

The action is in a special procedure that you write. This procedure is named after the .PRO or .SAV file that it's in: filename_define_buttons. For example, if the host file was called QWERTY.PRO or QWERTY.SAV then you would name your procedure: QWERTY_DEFINE_BUTTONS

The procedure has one positional parameter. Call it B. So:

```
pro qwerty_define_buttons, B
```

I think that stuff can come in via B sometimes but have never worked out how to deal with that. What I know is that you can define B from scratch with just your own menu additions and - if you do it right - these additions will find their way into the ENVI menu.

B is an array of structures, one element per item (node or leaf) of your menu additions. The structure definition is un-named, I think, and it looks like this:

```
{ parent:"", sibling:"", name:"", uvalue:"", event_handler:"",  
separator:0, display:0 }
```

Straight away, you can see that it's a lot like an entry in the envi.men file.

Anyway, for a given element in this structure array:

Set PARENT to the name of an existing node under which to add this entry, or set it to an empty string to add this entry as a top-level item in the ENVI menu.

Set SIBLING to the name of an existing node/leaf if you want this entry to appear just after it, or set it to an empty string to make this entry "go at the end".

Set NAME to what you would like to call this node/leaf in the ENVI menu. Leave the other members empty/zero if this entry is a node. If it is a leaf (a module invocation) then continue...

Set UVALUE to anything you like. (I don't use this item myself and don't know what it's for.)

Set EVENT_HANDLER to the name of your module's top-level procedure (the same one that you would have specified in envi.men)

I think that SEPARATOR gives you one of those menu partition things. I haven't used it.

I think that DISPLAY, if set non-zero, might add your item to the display menu rather than the main menu. I haven't used it - I've always left it zero to add to the main menu.

That's about it. You simply set up a structure array and return it.

When ENVI starts, it goes looking in SAVE_ADD, restoring or compiling modules that it finds there. For each module, it'll look for a ..._DEFINE_BUTTONS procedure and it'll run it if found.

What I don't know is how to get some interaction going if you have more than one module that's trying to add menus. i.e., If they share a common root node but it's a node that must be created, which one should have the honours? The trouble is that ENVI appears to discover files (in SAVE_ADD) according to some lower-level file-system thing rather than alphabetically, so you can't count on a discovery order. (I just try to avoid this situation by combining related modules into one .SAV file and having one DEFINE_BUTTONS procedure for the lot.)

HTH

Peter Mason

Subject: Re: ENVI menus, programmatically.

Posted by [Robert Moss](#) on Tue, 16 Mar 2004 22:25:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Peter Mason wrote:

> Robert Moss wrote:

>

>> Does anyone here know the trick to programmatically adding menu items
>> to ENVI? Yes, I have "googled" on this topic, and found one instance
>> that looked promising, but it doesn't really answer the question.
>> Clearly there is a way to do it... and it appears to involve creating
>> a procedure that has one parameter which is a data structure that
>> defines the menu structure. What happens inside this routine, or what
>> undocumented programs it may call, or what undocumented program call
>> it, is non-obvious. Any takers?
>>
>> Robert Moss, PhD
>> EOIR Technologies Inc.
>> robert.moss@nvl.DONOTSPAMTHEarmy.mil
>
>
> I wish that RSI would get on with it and document this interface as it is
> very useful. IMHO, they under-estimate the nuisance factor involved in the
> traditional route of editing the menu file. Even if you give
> copy-and-paste instructions to your users, it's still a nuisance as they
> have to find the envi.men file, overcome the geek factor, and so on. Most
> ENVI users can't be having with this sort of run-around. (Other sorts,
> sure, but not this one :-))
> Whereas.
> I sort-of found out how this interface works a good few years ago and use it
> routinely. *It has been stable for a number of ENVI versions.* In the
> interests of motivating some genuine documentation, here's what I know:
>
> The system only kicks in if your file (.PRO or .SAV) is in ENVI's SAVE_ADD
> subdirectory.
>
> The action is in a special procedure that you write. This procedure is
> named after the .PRO or .SAV file that it's in: filename_define_buttons.
> For example, if the host file was called QWERTY.PRO or QWERTY.SAV then you
> would name your procedure: QWERTY_DEFINE_BUTTONS
>
> The procedure has one positional parameter. Call it B. So:
> pro qwerty_define_buttons, B
> I think that stuff can come in via B sometimes but have never worked out how
> to deal with that. What I know is that you can define B from scratch with
> just your own menu additions and - if you do it right - these additions will
> find their way into the ENVI menu.
>
> B is an array of structures, one element per item (node or leaf) of your
> menu additions. The structure definition is un-named, I think, and it
> looks like this:
> { parent:"", sibling:"", name:"", uvalue:"", event_handler:"",
> separator:0, display:0 }
> Straight away, you can see that it's a lot like an entry in the envi.men
> file.

> Anyway, for a given element in this structure array:
> Set PARENT to the name of an existing node under which to add this entry, or
> set it to an empty string to add this entry as a top-level item in the ENVI
> menu.
> Set SIBLING to the name of an existing node/leaf if you want this entry to
> appear just after it, or set it to an empty string to make this entry "go at
> the end".
> Set NAME to what you would like to call this node/leaf in the ENVI menu.
> Leave the other members empty/zero if this entry is a node. If it is a
> leaf (a module invocation) then continue...
> Set UVALUE to anything you like. (I don't use this item myself and don't
> know what it's for.)
> Set EVENT_HANDLER to the name of your module's top-level procedure (the same
> one that you would have specified in envi.men)
> I think that SEPARATOR gives you one of those menu partition things. I
> haven't used it.
> I think that DISPLAY, if set non-zero, might add your item to the display
> menu rather than the main menu. I haven't used it - I've always left it
> zero to add to the main menu.
>
> That's about it. You simply set up a structure array and return it.
> When ENVI starts, it goes looking in SAVE_ADD, restoring or compiling
> modules that it finds there. For each module, it'll look for a
> ..._DEFINE_BUTTONS procedure and it'll run it if found.
> What I don't know is how to get some interaction going if you have more than
> one module that's trying to add menus. i.e., If they share a common root
> node but it's a node that must be created, which one should have the
> honours? The trouble is that ENVI appears to discover files (in SAVE_ADD)
> according to some lower-level file-system thing rather than alphabetically,
> so you can't count on a discovery order. (I just try to avoid this
> situation by combining related modules into one .SAV file and having one
> DEFINE_BUTTONS procedure for the lot.)
>
> HTH
> Peter Mason
>
>

Thanks, Peter. You've made my life a little easier.

R
