

Retrieving filenames with file dialogs via scripts and plugins

in BrainVoyager QX 2.2 and higher

Retrieving filenames with file dialogs via scripts and plugins.....	1
Get files names via scripting (and GUI).....	2
Get file names via plugin and GUI script.....	4
More information.....	6

Latest update: 31-10-2014

Get files names via scripting (and GUI)

1. In Qt Creator, add a push button and give this widget a name, for example "btnGetFiles" (see figure 1).

2. In the dialog initialisation function (`scriptObj.initDlg = function()`):

a. add a line to find the widget using the function `FindChild()`:

```
getFileButton = BrainVoyagerQX.FindChild("btnGetFiles");
```

b. add a line to connect the widget to a function, for example to the function

```
getNativeProcessingFiles():
```

```
getFileButton.clicked.connect(this, this.getNativeProcessingFiles);
```

3. In the top of the script, add a global variable that represents the widget (here: button) for example `"var getFileButton;"`.

4. In the function that is activated when clicking the button (here:

`getNativeProcessingFiles()`), add the line to present a file dialog, using the function `BrowseFile()`:

```
var projfilename = BrainVoyagerQX.BrowseFile("Please select the FMR project",  
"*.fmr");
```

where filetype is an extension string, for example `"*.fmr"`.

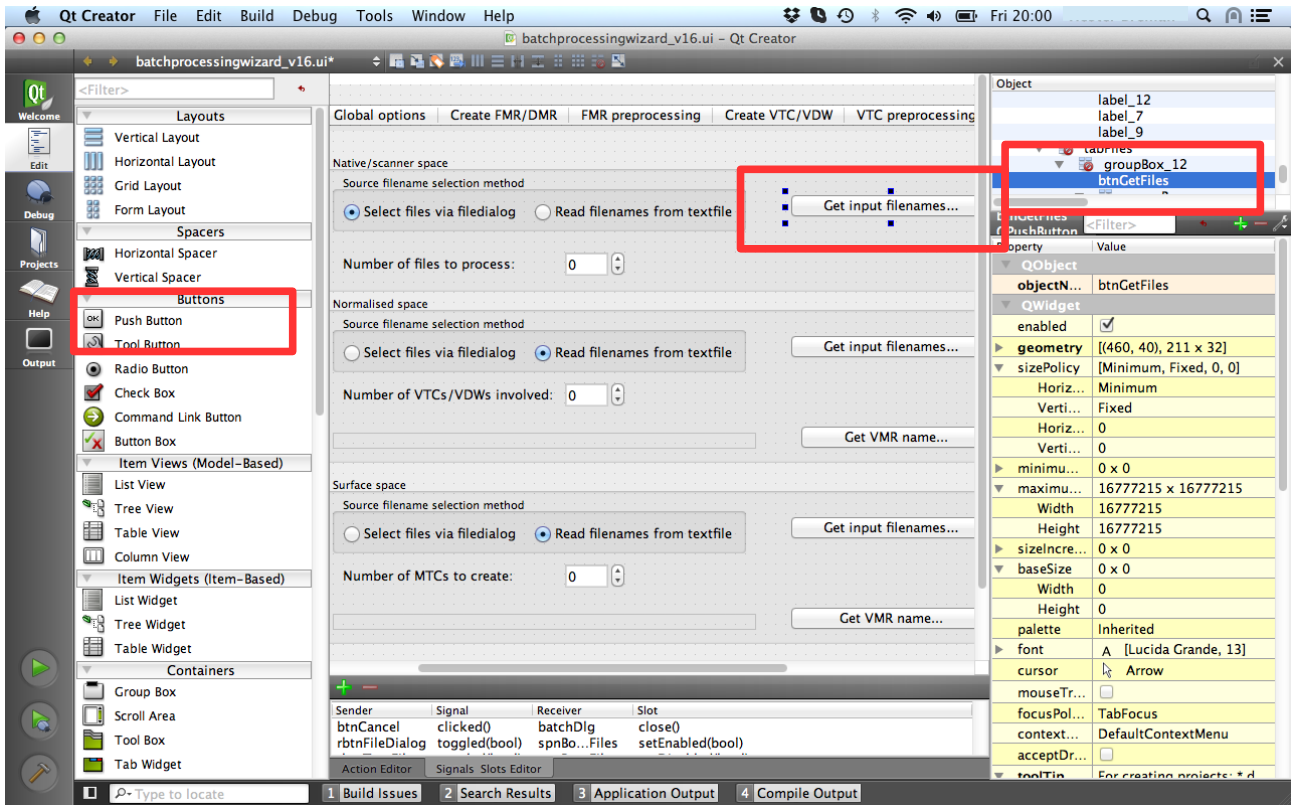


Figure 1: add a push button and give it a good name; remember the name.

```

530 spbZfFrom = BrainVoyagerQX.FindChild("spbZfFrom");
531 spbZto = BrainVoyagerQX.FindChild("spbZto");
532
533 // ~ widgets on Preprocess VTCs tab
534 rbtnVTCmm = BrainVoyagerQX.FindChild("rbtnVTCmm");
535 rbtnVTCvox = BrainVoyagerQX.FindChild("rbtnVTCvox");
536 dspbFWHMSpatialSmooth = BrainVoyagerQX.FindChild("dspbFWHMSpatialSmooth");
537 rbtnVTCHz = BrainVoyagerQX.FindChild("rbtnVTCHz");
538 rbtnVTCcyc = BrainVoyagerQX.FindChild("rbtnVTCcyc");
539 dspbVTCFWHMTHP = BrainVoyagerQX.FindChild("dspbVTCFWHMTHP");
540 rbtnVTCsec = BrainVoyagerQX.FindChild("rbtnVTCsec");
541 rbtnVTCDataPoints = BrainVoyagerQX.FindChild("rbtnVTCDataPoints");
542 dspbVTCFWHMTGS = BrainVoyagerQX.FindChild("dspbVTCFWHMTGS");
543
544 // ~ widgets on Create MTCs tab
545 dspbMTCFrom = BrainVoyagerQX.FindChild("dspbMTCFrom");
546 dspbMTCto = BrainVoyagerQX.FindChild("dspbMTCto");
547
548 // ~ widgets on files tab
549 // native/scanner space
550 getFileButton = BrainVoyagerQX.FindChild("btnGetFiles");
551 nrofilesBox = BrainVoyagerQX.FindChild("spnBoxNrOfFiles");
552 selectFilesNowBox = BrainVoyagerQX.FindChild("rbtnFileDialog");
553 selectTextFileBox = BrainVoyagerQX.FindChild("rbtnTextFile");

```

Figure 2: In the dialog initialisation function (`scriptObj.initDlg = function()`), add a line to find the widget using the function `FindChild()`.

Get file names via plugin and GUI script

Follow description 1-3 from above. Then:

4. Create a name and function in the plugin code to catch the command and parameter to get a file name, for example "Command" and "FileType":

```
char task_name[101];
qxGetStringParameter("Command", task_name);
char filetype[101];
qxGetStringParameter("FileType", filetype); // "*.fmr" or another
extension, or any file: "*.*"
```

5. In the GUI script (see Figure 3):

a. set the command and the parameter,

```
BrainVoyagerQX.SetPluginStringParameter("Command", "GetFileName");
BrainVoyagerQX.SetPluginStringParameter("FileType", filetype);
```

b. then invoke the plugin:

```
BrainVoyagerQX.ExecutePlugin();
```

6. In the plugin (see Figure 4):

a. First retrieve the command

```
qxGetStringParameter("Command", task_name);
```

b. Then retrieve the filetype (file extension) that needs to be retrieved:

```
qxGetStringParameter("FileType", filetype);
```

c. the filename is retrieved with the function qxGetOpenFileName(); the return value (integer) indicates whether or not a name was selected (1) or not (0):

```
int usrResp = qxGetOpenFileName(instruction, "", typeOfFiles, filename);
```

d. Communicate the filename to the GUI script:

```
qxSetStringParameter("Filename", filename);
```

7. In the GUI script, retrieve the filename via the function

GetPluginStringParameter():

```
var filename = BrainVoyagerQX.GetPluginStringParameter("Filename");
```

```

544
545 scriptObj.getFileName = function filetype {
546
547     var filename = "";
548     BrainVoyagerQX.SetPluginStringParameter("Command", "GetFileName");
549     BrainVoyagerQX.SetPluginStringParameter("FileType", filetype);
550     BrainVoyagerQX.ExecutePlugin();
551     if (this.noErrors()) {
552         filename = BrainVoyagerQX.GetPluginStringParameter("Filename");
553         BrainVoyagerQX.PrintToLog("Selected: " + filename);
554     }
555     return filename;
556 }
557

```

Figure 3: preparation work to send information to the plugin for retrieving the filename

```

// let's get some parameters from GUI script:
char task_name[101];
qxGetStringParameter("Command", task_name);
char filetype[101];
qxGetStringParameter("FileType", filetype); // should be including wildcard

if (!strcmp(task_name, "GetFileName")) {

    char *instruction      = "Please select the source file";
    char *defaultFolder   = new char[256];
    qxGetBrainVoyagerQXPath(defaultFolder);
    char *typeOfFiles     = filetype; // "BrainVoyager functional project (*.fmr)";
    char *filename        = new char[256];

    int usrResp = qxGetOpenFileName(instruction, "", typeOfFiles, filename);
    if (usrResp == 1) {
        qxSetStringParameter("Filename", filename);
        qxSetIntParameter("ErrorNr", 0);
    } else {
        qxSetIntParameter("ErrorNr", -1);
        qxSetStringParameter("ErrorMsg", "No file selected");
    }
}

```

Figure 4: retrieve the file name via `qxGetOpenFileName()` and send it back to the script via `qxSetStringParameter()`.

More information

For more information about the scripting functions, please consult the Scripting Reference Manual. This can be downloaded from our support site:

<http://support.brainvoyager.com/automation-aamp-development/46-writing-scripts/133-scripting-reference-guide.html>

and is installed with BrainVoyager in /BrainVoyager/GettingStartedGuides/.

The BrainVoyager plugin access functions are described in the Developer Guide. This is installed with BrainVoyager in /BrainVoyager/UsersGuide/.