

SuperApe

<UsingLibrary>

```
include;
https://monkeysaudio.com/;
```

SuperCodec

<UsingLibrary>

```
include;
superapi::supermultimedia;
superapi::superape;
superapi::superflac;
superapi::supermp3;
```

SuperColorDialog

<UsingLibrary>

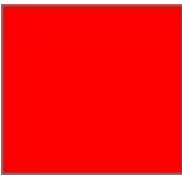
```
include;
superapi::superwindow;

public;
superapi::superwindow;
```

<SuperWidget>



```
clicked;
choose this color;
```



```
nousercontrol;
display current color;
```

```
userinput;
edit current color's red value;
display current color red value;
```

```
userinput;
edit current color's green value;
display current color green value;
```

```
userinput;
edit current color's blue value;
display current color blue value;
```

```
clicked;
change basic color;
```

```
nousercontrol;
display html color;
```



OK

Cancel

SuperCore

```
<UsingLibrary>
```

```
include; c/c++,qt library etc;
```

SuperFftw

```
<UsingLibrary>
```

```
include;
http://www.fftw.org/;
```

SuperFileDialog

<UsingLibrary>

```
include;  
superapi::supernotedialog;  
superapi::superwindow;
```

```
public;
superapi::superwindow;
```

<SuperWidget>


```
clicked;  
cd to last directory;
```



```
clicked;
refresh current directory;
```



```
clicked;
lock current directory to quick access;
```

D:/repos/QtWidget 6/build-SuperTools-Desktop_Qt_6_2_2_MinGW_64_bit-Release/release/_htmllibrary_

```
textActivated;
    if directory is vaild,refresh current folder;
    else show error dialog;
```

Quick access

```
itemclicked;
    if directory is vaild,refresh current folder;
    else show error dialog;
```

```
itemrightclicked;  
    show quickaccessitemmenu;  
    include "Open folder", "Unpin folder";
```

Windows 10 (C:)

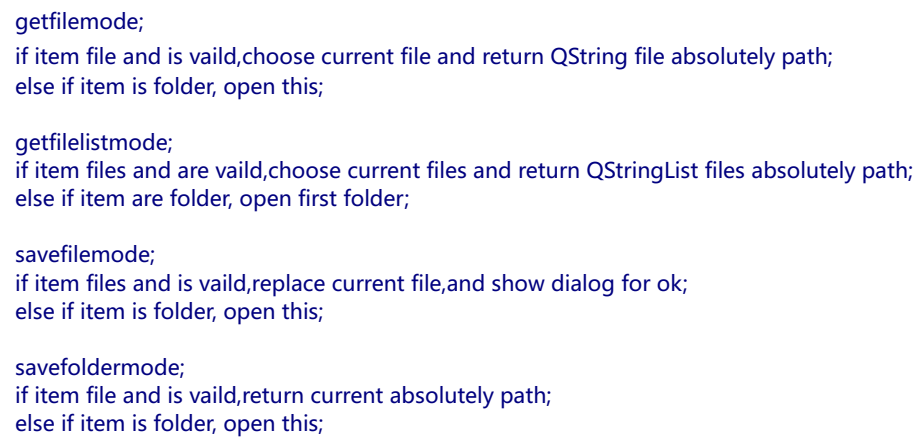
```
itemclicked;
    if drive is valid, refresh current folder;
    else show error dialog;
```



```
clicked;
add folder at current folder;
```



```
find and highlight file or folder at current folder;  
enable or disable ok* button;
```



```
getfilemode && getfilelistmode;
if item file and is vaild,show itemmenu;
include "Select file","Open file","Open path","Delete file";
else if item is folder,show foldermenu;
include "Open folder","No function","Open path","Delete folder"

savefilemode;
if item file and is vaild,show itemmenu;
include "Replace file","Open file","Open path","Delete file";
else if item is folder,show foldermenu;
include "Open folder","No function","Open path","Delete folder"

savefoldermode;
if item file and is vaild,show itemmenu;
include "Select path","Open file","Open path","Delete file";
else if item is folder,show foldermenu;
include "Open folder","No function","Open path","Delete folder"
```



filename:

```
sequenceDiagram
    participant userinput;
    participant filter:
    participant nousercontrol;
    participant Open
    participant clicked;
    participant Cancel

    userinput->>filter: getfilename && getfilemode && savefoldermode; disabled; savefilemode; if have text,enable save button; else disable save button;
    activate filter:
    filter:->>nousercontrol: *.png
    deactivate filter:
    activate nousercontrol;
    nousercontrol->>Open: getfilename && getfilelistmode && savefoldermode; disabled,display current file dialog's name filter; savefilemode; enable,display current file dialog's name filter;
    deactivate nousercontrol;
    activate Open
    Open->>clicked: Open
    deactivate Open
    activate clicked;
    clicked->>Cancel: clicked;
    deactivate clicked;
    activate Cancel
    Cancel->>Cancel: close this file dialog;
    deactivate Cancel
```

The diagram illustrates the sequence of events in a file dialog. It starts with a **userinput;** object sending a message to a **filter:** object. The message contains file-related actions and conditional logic for enabling or disabling the save button. The **filter:** object then sends a message to a **nousercontrol;** object, specifying the file filter as ***.png**. The **nousercontrol;** object sends a message to an **Open** button object. The **Open** button object sends a message to a **clicked;** object. Finally, the **clicked;** object sends a message to a **Cancel** button object, which then performs the action **close this file dialog;**.

SuperFileManager

<UsingLibrary>

```
include;
    superapi::supercore;
    superapi::superfiledialog;
    superapi::supernotedialog;
```

<SuperWidget>

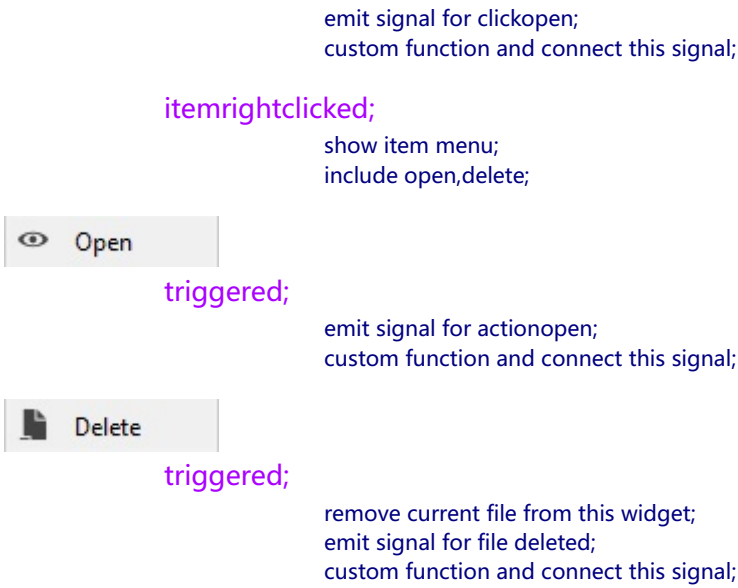
```

classDiagram
    class SuperFileManagerModule {
        clicked()
        clicked()
        clicked()
        textchanged()
        Name
        Path
        triggered()
        triggered()
        itemdoubleclicked()
    }
    class SuperFileManagerWidget {
        clicked()
        clicked()
        clicked()
        textchanged()
        headerrightclick()
        triggered()
        triggered()
        itemdoubleclicked()
    }
    SuperFileManagerModule --> SuperFileManagerWidget : clicked;
    SuperFileManagerWidget --> SuperFileManagerModule : clicked;
    SuperFileManagerModule --> SuperFileManagerWidget : clicked;
    SuperFileManagerWidget --> SuperFileManagerModule : textchanged;
    SuperFileManagerModule --> SuperFileManagerWidget : Name
    SuperFileManagerModule --> SuperFileManagerWidget : Path
    SuperFileManagerWidget --> SuperFileManagerModule : triggered;
    SuperFileManagerModule --> SuperFileManagerWidget : triggered;
    SuperFileManagerWidget --> SuperFileManagerModule : itemdoubleclicked;
  
```

The diagram illustrates the interaction between a **Super File Manager Module** and a **Super File Manager** widget. The module and widget are connected via a series of signals and slots, as well as properties.

- Module to Widget:**
 - `clicked;` (purple) → `display current file type as title;` (grey)
 - `clicked;` (purple) → `determined last file that opened;` (grey)
 - `clicked;` (purple) → `check all files are valid or not;` (grey)
 - `textchanged;` (purple) → `if current text is not empty, search current text at treewidget; else show all items;` (grey)
 - `Name` (yellow) and `Path` (yellow) → `headerrightclick;` (purple)
 - `triggered;` (purple) → `open super file dialog and select file or files; if files exist, skip this; else add files to treewidget;` (grey)
 - `triggered;` (purple) → `show super notedialog for make sure; if ok, clear all files; else close dialog and do nothing;` (grey)
- Widget to Module:**
 - `clicked;` (purple) → `determined last file that opened;` (grey)
 - `textchanged;` (purple) → `if current text is not empty, search current text at treewidget; else show all items;` (grey)
 - `headerrightclick;` (purple) → `show table menu; include import super files, delete all files;` (grey)
 - `triggered;` (purple) → `open super file dialog and select file or files; if files exist, skip this; else add files to treewidget;` (grey)
 - `triggered;` (purple) → `show super notedialog for make sure; if ok, clear all files; else close dialog and do nothing;` (grey)
 - `itemdoubleclicked;` (purple) → `show super notedialog for make sure; if ok, clear all files; else close dialog and do nothing;` (grey)

The diagram also shows the visual components of the Super File Manager widget, including a search bar, a table with columns **Name** and **Path**, and two buttons: **Import Super Files** and **Delete All Files**. The widget is currently displaying a file named `_hitbricks_super` with a path of `D:/repos/QtWidget 6/`.



SuperGroupBox

<UsingLibrary>

```
include;
qt groupbox assembly;
```

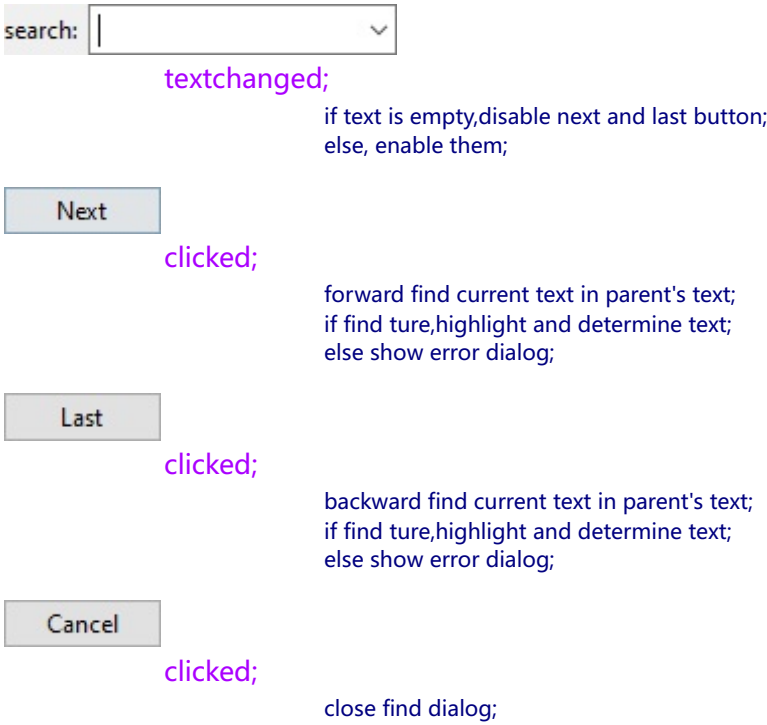
SuperFindDialog

<UsingLibrary>

```
include;
superapi::superwindow;
superapi::supernotedialog;

public;
superapi::superwindow;
```

<SuperWidget>



SuperFlac

<UsingLibrary>

```
include;
https://github.com/xiph/flac;
```

SuperFontDialog

<UsingLibrary>

```
include;
```

```
superapi::superwindow;

public;
superapi::superwindow;

<SuperWidget>

    here for search font...

    textchanged;
        if text is empty,display all font families;
        else, display contain text's families;

    Microsoft YaHei

    itemdoubleclicked;
        return current font families;

    itempressed;
        preview font families;
```

SuperTools

```
nousercontrol;
    display families style;
```

SuperHook

```
<UsingLibrary>

include;
    mircosoft windows api;
```

SuperLogger

```
<UsingLibrary>

include;
    superapi::supercore;

<SuperWidget>

    Clear Browser

    triggered;
        clear textbrowser;

    here will display wasapibox programe run logs!!!

    nousercontrol;
        display programe run logs;
```

SuperMp3

```
<UsingLibrary>

include;
    http://www.mpg123.de/;
```

SuperMultiMedia

```
<UsingLibrary>

include;
    superapi::supercore;
    qt multimedia assembly;
```

SuperNetwork

```
<UsingLibrary>

include;
    qt network assembly;
```

SuperNoteDialog

<UsingLibrary>

```
include;
    superapi::superwindow;

public;
    superapi::superwindow;
```

<SuperWidget>

Save Success!!

nousercontrol;
 display message;

Cancel

clicked;
 close super notedialog;

OK

clicked;
 close super notedialog and emit signal;

SuperPlot

<UsingLibrary>

```
include;
    https://www.qcustomplot.com/index.php;
```

SuperProgressBarDialog

<UsingLibrary>

```
include;
    superapi::superwindow;

public;
    superapi::superwindow;
```

<SuperWidget>

nousercontrol;
 display current progress;

wait message...

nousercontrol;
 display current message;

Stop

triggered;
 emit stop signal;
 custom function and connect this signal;

SuperSender

<UsingLibrary>

```
include;
    superapi::supernotedialog;
```

<SuperWidget>

Input line 00

nousercontrol;
 display current super sender index;

help

textchanged;
 if text is empty,disable send button;
 else if text contain <split> ,show error dialog and clear text;

Send

Is Cyc: ☐

1000

else ,enable send button;

clicked;

emit signal include lineedit's text;
custom function and connect this signal;

statechanged;

change send mode,cycle or not;

userinput;

change cycle interval,unit(ms);

SuperSerial

<UsingLibrary>

include;

superapi::supercore;
qt serialport assembly;

SuperTab

<UsingLibrary>

include;

qt widget assembly;

SuperTabWidget

<UsingLibrary>

include;

qt tabwidget assembly;

SuperWindow

<UsingLibrary>

include;

superapi::supercore;

public;

qt qwidget;

<SuperWidget>

SuperTools 6.20220128.018

nousercontrol;

display super window's title;

-

clicked;

set super windows minisize;

^

clicked;

set super windows maxisize;

<

clicked;

set super windows normal;

X

clicked;

close super windows;

clicked;

if Icon mode enabled,exec themes menu;
include original,color,heartofiron,goertek;



triggered;

change background title as original;
change background color as (16,81,142) ;
change font color as (255,255,255);



triggered;

open color dialog;
change background title as color;
change background color as selected ;
change font color as concolor;



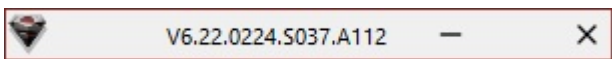
triggered;

change background title as heartofiron;
change background color as (121,35,32) ;
change font color as (255,255,255);



triggered;

change background title as goertek;
change background color as (69,86,17) ;
change font color as (255,255,255);



doubleclicked;

if enable maximum mode,change size as normal or maxisize;

mousemove;

if mouse pressed,move super window;



mouseonborder;

if enable resize mode;
A cursor used for elements that are used to vertically resize top-level windows.



mouseonborder;

if enable resize mode;
A cursor used for elements that are used to horizontally resize top-level windows.



mouseonborder;

if enable resize mode;
A cursor used for elements that are used to diagonally resize top-level windows at their top-left and bottom-right corners

Version Logs

<20220319 Version 1.21>

update supercore api;
optization linespace,arrage function;
fix superwindow api bugs;
fix disable close button function bugs,add disable button function;

<20220316 Version 1.20>

update superfinddialog api;
connect find next text function with combobox activate signal;

<20220313 Version 1.19>

update superhook api;
add win32 user.lib;

<20220309 Version 1.18>

update superfilemanager api;
import file add determined function;

<20220306 Version 1.17>

update superfiledialog api;
use superfilesystemmodel public qfilesystemmodel;
expand default path at init;

<20220304 Version 1.16>

remove supervisa api;
remove superwidgettimer api;

<20220301 Version 1.15>

update supertab api;
replace creat cache file at super.dll as at supertab api;
add cache function,include creat file,creat folder;

<20220228 Version 1.14>

update superwindow api;
add title icon mouse enter and leave function;

<20220225 Version 1.13>

update superwindow api;
add signal after show function;

<20220222 Version 1.12>

update supercore api;
optization code;

<20220215 Version 1.11>

update superwindow api;
update change window status function;

<20220215 Version 1.10>

update superfilemanager api;
add search text parse id item column;
update superfiledialog api;
replace scroll mode item as pixel;

<20220214 Version 1.09>

update superfilemanager api;
remove timer for emit logs,add reset index function when delete files;

<20220212 Version 1.08>

update superlogger api;
add transfercodec function,add openfilepath function;
update superfilemanager api;
add open file path function,add check file vaild function,add id for files;

<20220211 Version 1.07>

update supercore api;
update supercore::runcommand function;
update superfiledialog api;
fix superfiledialog get filename or foldername function bugs;
update superwindow::addmenu function;
add addseparator parameter;

<20220209 Version 1.06>

update supertab api;
add readme function;
update superlogger api;
add clear textbrowser function,displaylog function;
update supercore api;
remove textbrowser function,displaylog function;

<20220209 Version 1.05>

update supermultimedia api;
replace write data at run function with emit signal at main thread;

<20220207 Version 1.04>

add supertab api;
add supertabwidget api;
add supergroupbox api;

<20220205 Version 1.03>

supermultimedia playwav class add channel parameter;

<20220130 Version 1.02>

supermultimedia playwav class add timer for determine filepos;

<20220129 Version 1.01>

first version released;
update super window's icon when at deactivate status;

<20220128 Version 1.00>

first version released;