

'),o.close()}("https://assets.zendesk.com/embeddable_framework/main.js","numxl.zendesk.com"); /*]]>*/

Logging

Last Modified on 03/24/2016 6:24 pm CDT

- [C/C++](#)
- [.Net](#)

The SFLOG module provide the logging system functionality to NumXL SDK, and can be used as well by your application. The logging subsystem is implemented after the common "log4j" library, and support similar configuration and functionality.

Furthermore, the library is thread-safe, and if more than one process try to write to the same file, the system will create a new file for the second process to avoid collision.

Configuration file

In the main configuration file (e.g. "MyApp.conf"), locate the section for logging (i.e. [LOG]), and edit the different entries to suite your exact needs (e.g. Layout, level, etc.). Here is an example:

```
# LOG LEVELS: OFF=0, TRACE=1, DEBUG=2, INFO=3, WARN=4, ERROR=5, FATAL=6,
ALL= 7
[LOG]
FILE = MyApp.log
LEVEL = 7
BACKUPFILES = 7
MAXSIZE = 512000
COMPRESSION = false
LAYOUT = "%d %5p [%t] - %F(%L) - %c - %m%n"
```

Logging Level

E
n
a
b
l
e
a
l
l
l
o
g
g
i
n
g
m
e
s
s
a
g
e
s
(
B
F
L
O
G
G
d
A
L
l
g
,
t
r
a
c
e
,
w
a
r
n
i
n
g
a
n
d
e
r
r
o
r
)

E
n
a
b
l
e
t
r
a
c
c
e
l
e
v
e
l
l
o
g
g
i
n
g
:
t
r
a
c
c
,
d
e
b
S
F
L
D
G
f
T
R
W
E
n
i
n
g
,
e
r
o
r
a
n
d
f
a
t
a
l
e
r
r
o
r
m
e
s
s
a
g
e
s
.

E
n
a
b
l
e
d
e
b
u
g
l
e
v
e
l
l
o
g
g
i
n
g
:
d
e
b
u
g
,
B
E
L
O
G
2
w
D
E
B
U
G
,
e
r
r
o
r
a
n
d
f
a
t
a
l
e
r
r
o
r
m
e
s
s
a
g
e
s
.

E
n
a
b
l
e
i
n
f
o
r
m
a
t
i
o
n
l
e
v
e
l
l
o
g
g
i
n
g
:
i
S
P
E
C
I
F
I
C
A
T
I
O
N
S
G
W
A
I
N
I
N
F
R
O
N
g
,
e
r
r
o
r
,
a
n
d
f
a
t
a
l
e
r
r
o
r
m
e
s
s
a
g
e
s
.

E
n
a
b
l
e
w
a
r
r
n
i
n
g
l
e
v
e
l
l
o
g
g
i
n
g
:
S
W
F
L
O
G
G
i
n
g
:
W
A
R
R
o
r
r
a
n
d
f
a
t
a
l
l
e
r
r
o
r
m
e
s
s
a
g
e
s
.

E
n
a
b
l
e
e
r
r
o
r
l
e
v
e
l
l
o
g
g
i
S
P
E
C
I
F
I
C
A
T
I
O
N
S
F
O
R
R
E
C
O
R
D
f
a
t
a
l
e
r
r
o
r
m
e
s
s
a
g
e
s
.

E
n
a
b
l
e
f
a
t
a
l
O
S
F
C
L
O
G
6
F
C
A
P
I
A
L
L
e
v
e
l
l
o
g
g
i
n
g
.
D
i
s
a
b
l
e
a
B
F
L
O
7
G
D
E
F
m
e
s
s
a
g
e
s

Requirements


```
S
F
HL
eo
ag
dg
ee
rr
.
H

S
LF
iL
bO
rG
a.
rL
yI
B

S
F
L
DO
LG
L.
D
L
L
```

Example

```
#include <iostream>
#include "SFMacros.h"
#include "SFLogger.h"

using namespace std;

#define LOG_TRACE(x) \
    SFLOG_logMsg(SFLOG_TRACE,
    __FILE__, __FUNCTION__, __FUNCSIG__, __LINE__, x);
```

```

#define LOG_ERROR(x) \
    SFLOG_logMsg(SFLOG_ERROR,
    __FILE__, __FUNCTION__, __FUNCSIG__, __LINE__, x);

int main(){
    int nRet= NDK_FAILED;

    string szAppName="MyLogExample";
    string szPath = "C:\\temp";

    nRet = SFLOG_Init(szAppName.c_str(), // Use MyLogExample.conf file,
and name the log file as MyL0gExample.log
                    szPath.c_str()      // Use C:\temp folder to create
the log          file(s)
                    );
    if(nRet < NDK_SUCCESS){
        std::cerr << "Initializing the logging facility failed, exit" <<
std::endl;
        exit(-1);
    }
    SFLOG_TRACE("SF Logging facilty has been initialized successfully");
    .....

    // Change the logging level to Error
    nRet = SFLOG_SETLEVEL(SFLOG_ERROR);
    if( nRet < NDK_SUCCESS){
        SFLOG_ERROR("Changing logging level failed");
    }
    // Close the log file and shutdown the logging
    nRet = SFLOG_Shutdown();

    exit(0);
}

```

The NumXLAPI.SFLOG wrapper class provides the logging system functionality to NumXL SDK, and can be used as well by your application. The logging subsystem is implemented after the common "log4j" library, and support similar configuration and functionality.

Furthermore, the library is thread-safe, and if more than one process try to write to the same file, the system will create a new file for the second process to avoid collision.

Configuration file

In the main configuration file (e.g. "MyApp.conf"), locate the section for logging (i.e. [LOG]), and edit the different entries to suite your exact needs (e.g. Layout, level, etc.). Here is an example:

```
# LOG LEVELS: OFF=0, TRACE=1, DEBUG=2, INFO=3, WARN=4, ERROR=5, FATAL=6,
ALL= 7
[LOG]
FILE = MyApp.log
LEVEL = 7
BACKUPFILES = 7
MAXSIZE = 512000
COMPRESSION = false
LAYOUT = "%d %5p [%t] - %F(%L) - %c - %m%n"
```

Logging Level

```
namespace NumXLAPI{
    . . .
    public enum SFLOG_LEVEL
    {
        SFLOG_ALL=0,
        SFLOG_TRACE=1,
        SFLOG_DEBUG=2,
        SFLOG_INFO=3,
        SFLOG_WARN=4,
        SFLOG_ERROR=5,
        SFLOG_FATAL=6,
        SFLOG_OFF=7
    }
}
```

See Also

[template("related")]