

minispy.c

DriverEntry

Vista

```

MiniSpyData.PFItSetTransactionContext = (PFLT_SET_TRANSACTION_CONTEXT) FltGetRoutineAddress( "FItSetTransactionContext" );
MiniSpyData.PFItGetTransactionContext = (PFLT_GET_TRANSACTION_CONTEXT) FltGetRoutineAddress( "FItGetTransactionContext" );
MiniSpyData.PFItEnlistInTransaction = (PFLT_ENLIST_IN_TRANSACTION) FltGetRoutineAddress( "FItEnlistInTransaction" );

```

FItRegisterFilter

&FilterRegistration
Callbacks

②SpyPreOperationCallback

SpyNewRecord

```
recordList = SpyNewRecord();
```

FItGetFileNameInformation
FItParseFileNameInformation
FItReleaseFileNameInformation
FItGetFileNameInformation
SpySetRecordName
FItReleaseFileNameInformation
SpyLogPreOperationData

RecordListに登録する

SpyPostOperationCallback
SpyEnlistInTransaction

```
SpyLog( recordList );  
SpyLogTransactionNotify
```

②SpyPostOperationCallback

SpyLogPostOperationData
SpyLog(recordList);

Contexts

SpyDeleteTxfContext

SpyFilterUnload
SpyQueryTeardown
Vista

マニュアルでVolumeをdetachした場合

SpyKtmNotificationCallback

```
SpyLog( recordList );
```

&MiniSpyData.Filter

FItBuildDefaultSecurityDescriptor
FItCreateCommunicationPort
SpyConnect
SpyDisconnect
FItCloseClientPort
SpyMessage
Switch

GetMiniSpyLog

④SpyGetLog

```
while (!IsListEmpty( &MiniSpyData.OutputBufferList ) && (OutputBufferLength > 0)) {  
pList = RemoveHeadList( &MiniSpyData.OutputBufferList );  
InsertHeadList  
SpyFreeRecord( pRecordList );  
}
```

GetMiniSpyVersion

FItFreeSecurityDescriptor
FItStartFiltering

mspyUser.c

main

FilterConnectCommunicationPort

CreateThread

mspyLog.C!RetrieveLogRecords
③FilterSendMessage(GetMiniSpyLog,buffer)

ScreenDump
FileDump

while (inputChar = (CHAR)getchar()) {
InterpretCommand

switch (parm[1]) {
case 'a':
case 'A':

```
①FilterAttach( MINISPY_NAME,(PWSTR)buffer)
```

case 'd':
case 'D':

```
FilterDetach( MINISPY_NAME,(PWSTR)buffer,instanceString );
```

case 'l':
case 'L':

```
ListDevices();  
FilterVolumeFindFirst
```

case 's':
case 'S':

```
Context->NextLogToScreen = !Context->NextLogToScreen;
```

case 'f':
case 'F':

```
Context->OutputFile = fopen( parm, "w" );
```