

ADos

COLLABORATORS

	<i>TITLE :</i> ADos		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 13, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ADos	1
1.1	AmigaTalk to AmigaDOS Help:	1
1.2	SAFE AmigaDOS Methods:	2
1.3	UNSAFE AmigaDOS Methods:	3
1.4	DANGEROUS AmigaDOS Methods:	4
1.5	VERY DANGEROUS AmigaDOS Methods:	5
1.6	ADos Full Index:	6

Chapter 1

ADos

1.1 AmigaTalk to AmigaDOS Help:

The AmigaTalk interface to the AmigaDOS functions has been divided up into four classifications:

SAFE - Incorrect parameters will not cause major problems.

UNSAFE - Ensure that parameters & methods that are marked UNSAFE are checked for correct return values.

DANGEROUS - The Methods that use DANGEROUS AmigaDOS functions could cause loss of data or could freeze the Operating System!

VERY DANGEROUS - The Methods that use VERY DANGEROUS AmigaDOS functions could cause a major loss of data (example: the Format function), or could lock up your PC!

WARNING: Know what you're doing when using the AmigaTalk-to-AmigaDOS interface (If you destroy your HardDisk contents, you were warned)!

Each Method is first described by the AmigaDOS documentation for the underlying AmigaDOS function, then by the AmigaTalk Method description. Hopefully, this will force you to read what each function does to the Operating System of your Amiga first, before trying to do something potentially destructive to your Amiga-PC.

None of the Methods here will present a Requester re: your sanity (unless the underlying AmigaDOS function does so), so you will have no feedback about what your AmigaTalk program is doing, unless you create your own sub-classes that hold your hand better. My intent was to implement the functionality of AmigaDOS as closely as possible, so if you do not like something about these classes, either re-write AmigaDOS or create your own (more bullet-proof)

AmigaTalk classes!

Check them out:

SAFE	Methods
UNSAFE	Methods
DANGEROUS	Methods
VERY DANGEROUS	Methods
- Here be Dragons!!	

1.2 SAFE AmigaDOS Methods:

The functions listed here & used by AmigaTalk have been deemed the least harmful of the AmigaDOS functions. This is based on my judgement only, but here is how I arrived at this: The functions determined to be safe are mainly for gathering information from AmigaDOS. Those that actually change things are easily corrected by the User (for example: If you use `setComment: commentString onFile: fileName` & you used the wrong `commentString`, you can always re-do the Method with the correct `comment`, or correct it using a file utility, such as `DirOpus` or `DiskMaster II`). Where it made sense to do so, the arguments the User supplies these functions/Methods are also checked for valid ranges or values, so even if you pass in a NULL pointer, AmigaTalk should short-circuit your attempt to kill your system (I hope!).

SAFE AmigaDOS Functions/AmigaTalk Methods:

```
waitForChar
vPrintf
vFPrintf
unGetC
strToLong
strToDate
splitName
setProtection
setPrompt
setIoErr
setFileDate
setComment
sameLock
sameDevice
readLink
readItem
readArgs
readFile
putStr
printFault
pathPart
parentOfFH
```

```
parentDir
maxCli
matchNext
matchFirst
matchEnd
isInteractive
isFileSystem
ioErr
getVar
getPrompt
getProgramName
getProgramDir
getFileSysTask
getDeviceProc
getCurrentDirName
getConsoleTask
getArgStr
fPutS
fPutC
findVar
findCliProc
fGetS
fGetC
fault
endNotify
delay
dateToStr
currentDir
compareDates
cliPointer
addBuffers
abortPacket
```

1.3 UNSAFE AmigaDOS Methods:

It's advisable that the User of these AmigaTalk Methods check their work before usage.

This classification is based on my judgement only, but here is how I arrived at this: The functions determined to be Unsafe can change file Locks, traverse a directory, or change something (such as an Assignment) that might not be easily found or corrected.

Where it made sense to do so, the arguments the User supplies these functions/Methods are also checked for valid ranges or values, so even if you pass in a NULL pointer, AmigaTalk should short-circuit your attempt to kill your system (I hope!).

UNSAFE AmigaDOS Functions/AmigaTalk Methods:

```
writeChars
vFWritef
unLockRecords
unLockRecord
```

```
unLockDosList
unLock
startNotify
setVar
setProgramName
setProgramDir
setOwner
setMode
setCurrentDirName
rename
relabel
parsePatternNoCase
parsePattern
output
openFromLock
openFile
nextDosEntry
nameFromLock
nameFromFH
matchPatternNoCase
matchPattern
makeLink
makeDosEntry
lockRecords
lockRecord
lockDosList
lock
input
infoDisk
fRead
flushFH
findSegment
findDosEntry
findArg
exNext
execute
examineFH
examine
exAllEnd
exAll
dupLockFromFH
dupLock
dateStamp
closeFile
checkSignal
changeMode
assignPath
assignLock
assignLate
assignAdd
addPart
```

1.4 DANGEROUS AmigaDOS Methods:

WARNING: Improper usage of these Methods will (at the bare minimum), result in the Operating System hanging up, which could result in loss of data. This is the least of what could happen! Turn back now!

DANGEROUS AmigaDOS Functions/AmigaTalk Methods:

```
writeFile
setVBuf
setFileSize
setArgStr
selectOutput
selectInput
seekFile
runCommand
replyPkt
inhibit
fWrite
freeDosObject
freeDosEntry
freeDeviceProc
freeArgs
exitProgram          -- Avoid like the plague!
deviceProc
deleteVar
createNewProc
cliInitRun
cliInitNewcli
attemptLockDosList
allocDosObject
addDosEntry
```

1.5 VERY DANGEROUS AmigaDOS Methods:

WARNING: Improper usage of these Methods will (at the bare minimum), result in the Operating System hanging up, which could result in loss of data. This is the least of what could happen! Turn back now!

VERY DANGEROUS AmigaDOS Functions/AmigaTalk Methods:

```
waitPkt
unLoadSeg
systemTagList
setFileSysTask
setConsoleTask
sendPkt
remSegment
remDosEntry
remAssignList
newLoadSeg
loadSeg
internalUnLoadSeg
internalLoadSeg
format              -- Are you out of your tree??
doPacket
deleteFile
```


addSegment

1.6 ADos Full Index:

SAFE AmigaDOS Functions/AmigaTalk Methods:

waitForChar
vPrintf
vFPrintf
unGetC
strToLong
strToDate
splitName
setProtection
setPrompt
setIoErr
setFileDate
setComment
sameLock
sameDevice
readLink
readItem
readArgs
readFile
putStr
printFault
pathPart
parentOfFH
parentDir
maxCli
matchNext
matchFirst
matchEnd
isInteractive
isFileSystem
ioErr
getVar
getPrompt
getProgramName
getProgramDir
getFileSysTask
getDeviceProc
getCurrentDirName
getConsoleTask
getArgStr
fPutS
fPutC
findVar
findCliProc
filePart
fGetS
fGetC
fault
endNotify
delay
dateToStr

currentDir
compareDates
cliPointer
addBuffers
abortPacket

UNSAFE AmigaDOS Functions/AmigaTalk Methods:

writeChars
vFwritef
unLockRecords
unLockRecord
unLockDosList
unLock
startNotify
setVar
setProgramName
setProgramDir
setOwner
setMode
setCurrentDirName
rename
relabel
parsePatternNoCase
parsePattern
output
openFromLock
openFile
nextDosEntry
nameFromLock
nameFromFH
matchPatternNoCase
matchPattern
makeLink
makeDosEntry
lockRecords
lockRecord
lockDosList
lock
input
infoDisk
fRead
flushFH
findSegment
findDosEntry
findArg
exNext
execute
examineFH
examine
exAllEnd
exAll
dupLockFromFH
dupLock
dateStamp
closeFile
checkSignal

changeMode
assignPath
assignLock
assignLate
assignAdd
addPart

DANGEROUS AmigaDOS Functions/AmigaTalk Methods:

writeFile
setVBuf
setFileSize
setArgStr
selectOutput
selectInput
seekFile
runCommand
replyPkt
inhibit
fWrite
freeDosObject
freeDosEntry
freeDeviceProc
freeArgs
exitProgram -- Avoid like the plague!
deviceProc
deleteVar
createNewProc
cliInitRun
cliInitNewcli
attemptLockDosList
allocDosObject
addDosEntry

VERY DANGEROUS AmigaDOS Functions/AmigaTalk Methods:

waitPkt
unLoadSeg
systemTagList
setFileSysTask
setConsoleTask
sendPkt
remSegment
remDosEntry
remAssignList
newLoadSeg
loadSeg
internalUnLoadSeg
internalLoadSeg
format -- Are you out of your tree??
doPacket
deleteFile
addSegment
