

COWON S9_FS Command



REVISION HISTORY

Revision	Date	Description
1.1	2009-05-15	<ul style="list-style-type: none"> - Revised chapters <ul style="list-style-type: none"> 3.6.11 GetAudSampleRate 3.11.12 GetEtcTASValue 5.1.1 EtcBrwSetInitialization 5.4.6 EtcUsrSetMainmenu - Added Chapters <ul style="list-style-type: none"> 2.3 Video Mode Key FS Command Group 3.7 Video Mode Get Parameter FS Command Group 4.7 Video Mode Set Parameter FS Command Group 5.5 Notepad FS Command Group 3.2.8 GetDisWallpaper 3.6.20 GetAudPitchControl 3.11.14 GetEtcTASValueX 3.11.15 GetEtcTASValueY 3.11.16 GetEtcTASValueZ 4.2.8 SetDisWallpaper 4.2.9 SetDisLCDAutoOffReset 4.4.6 SetBTHPinCode 4.11.8 SetEtcWallpaper 5.1.10 EtcBrwGetType 5.2.2 EtcModSetWidgetMode

S9
©WON

CONTENTS

1 FS Command

- 1.1 FS Command Summary
- 1.2 Key FS Command
- 1.3 Get/Set Parameter FS Command
- 1.4 Etc FS Command

2 Key FS Command

- 2.1 Common Mode Key FS Command
 - 2.1.1 KeyComPlus
 - 2.1.2 KeyComMinus

2.2 Music Mode Key FS Command

- 2.2.1 KeyAudPlay
- 2.2.2 KeyAudPause
- 2.2.3 KeyAudStop
- 2.2.4 KeyAudShortFF
- 2.2.5 KeyAudShortREW
- 2.2.6 KeyAudLongFF
- 2.2.7 KeyAudLongREW
- 2.2.8 KeyAudDirectSeek
- 2.2.9 KeyAudBookmark
- 2.2.10 KeyAudFavorite
- 2.2.11 KeyAudABMode

2.3 Video Mode Key FS Command

- 2.3.1 KeyVidInit
- 2.3.2 KeyVidDeinit
- 2.3.3 KeyVidOpen
- 2.3.4 KeyVidPlay
- 2.3.5 KeyVidPause
- 2.3.6 KeyVidLongREW
- 2.3.7 KeyVidLongFF
- 2.3.8 KeyVidDirectSeek
- 2.3.9 KeyVidLongKeyRelease
- 2.3.10 KeyVidBookmark
- 2.3.11 KeyVidFavorite
- 2.3.12 KeyVidCapture
- 2.3.13 KeyVidMakeScene
- 2.3.14 KeyVidScenelnit

2.4 FM Radio Mode Key FS Command

- 2.4.1 KeyFmrRecStart
- 2.4.2 KeyFmrRecStop
- 2.4.3 KeyFmrShortFF
- 2.4.4 KeyFmrShortREW
- 2.4.5 KeyFmrLongFF
- 2.4.6 KeyFmrLongREW

2.5 Mobile TV(DMB/DAB) Mode Key FS Command(Domestic Version Only)

- 2.5.1 KeyMTVInit
- 2.5.2 KeyMTVDeinit

- 2.5.3 KeyMTVStop
- 2.5.4 KeyMTVSelService
- 2.5.5 KeyMTVAutoScan
- 2.5.6 KeyMTVScanStop
- 2.5.7 KeyMTVRec

2.6 Record Mode Key FS Command

- 2.6.1 KeyRecStart
- 2.6.2 KeyRecStop
- 2.6.3 KeyRecPause
- 2.6.4 KeyRecPlay

3 Get Parameter FS Command

3.1 JetEffect 2.0 Get Parameter FS Command

- 3.1.1 GetJetEffectIndex
- 3.1.2 GetJetUserIndex
- 3.1.3 GetJetUserData
- 3.1.4 GetJetUserBBE
- 3.1.5 GetJetUserM3B
- 3.1.6 GetJetUser3DS
- 3.1.7 GetJetUserMPE
- 3.1.8 GetJetUserSTE
- 3.1.9 GetJetUserEQBandGain
- 3.1.10 GetJetUserEQCenterFreq
- 3.1.11 GetJetUserEQBandWidth

3.2 Display Get Parameter FS Command

- 3.2.1 GetDisLanguage
- 3.2.2 GetDisFont
- 3.2.3 GetDisPlayTime
- 3.2.4 GetDisLyrics
- 3.2.5 GetDisLCDAutoOff
- 3.2.6 GetDisBrightness
- 3.2.7 GetDisTVOutMode
- 3.2.8 GetDisWallpaper

3.3 Timer Get Parameter FS Command

- 3.3.1 GetTimAlarmDuration
- 3.3.2 GetTimAlarmRepeat
- 3.3.3 GetTimAlarmMode
- 3.3.4 GetTimAlarmTime
- 3.3.5 GetTimAlarmState
- 3.3.6 GetTim24HDisplay
- 3.3.7 GetTimSleep
- 3.3.8 GetTimAutoOff

3.4 Bluetooth Get Parameter FS Command

- 3.4.1 GetBTHState
- 3.4.2 GetBTHConnect
- 3.4.3 GetBTHScan
- 3.4.4 GetBTHDeviceList
- 3.4.5 GetBTHPinCode

3.5 System Get Parameter FS Command

- 3.5.1 GetSysCtrlHoldState
- 3.5.2 GetSysPan
- 3.5.3 GetSysUSBMode
- 3.5.4 GetSysHoldKey
- 3.5.5 GetSysBattery
- 3.5.6 GetSysSleepMode
- 3.5.7 GetSysBeepMode
- 3.5.8 GetSysTotalSize
- 3.5.9 GetSysFreeSize
- 3.5.10 GetSysVersion

3.6 Music Mode Get Parameter FS Command

- 3.6.1 GetAudSkipLength
- 3.6.2 GetAudScanSpeed
- 3.6.3 GetAudResume
- 3.6.4 GetAudPSpeed
- 3.6.5 GetAudFadeln
- 3.6.6 GetAudRepeat
- 3.6.7 GetAudShuffle
- 3.6.8 GetAudBoundary
- 3.6.9 GetAudABMode
- 3.6.10 GetAudBitRate
- 3.6.11 GetAudSampleRate
- 3.6.12 GetAudCodec
- 3.6.13 GetAudPlayTime
- 3.6.14 GetAudTotalTime
- 3.6.15 GetAudAlbum
- 3.6.16 GetAudArtist
- 3.6.17 GetAudTitle
- 3.6.18 GetAudAlbumArtTotalNum
- 3.6.19 GetAudLDBType
- 3.6.20 GetAudPitchControl

3.7 Video Mode Get Parameter FS Command

- 3.7.1 GetVidWidth
- 3.7.2 GetVidHeight
- 3.7.3 GetVidFrameRate
- 3.7.4 GetVidSampleRate
- 3.7.5 GetVidBitRate
- 3.7.6 GetVidState
- 3.7.7 GetVidTVOutEnable
- 3.7.8 GetVidStartMode
- 3.7.9 GetVidOpenState
- 3.7.10 GetVidScene
- 3.7.11 GetVidSceneTime
- 3.7.12 GetVidSkipLength
- 3.7.13 GetVidResume
- 3.7.14 GetVidPSpeed
- 3.7.15 GetVidCaption

- 3.7.16 GetVidRepeat
- 3.7.17 GetVidABMode
- 3.7.18 GetVidPlayTime
- 3.7.19 GetVidPlayTimeMS
- 3.7.20 GetVidTotalTime
- 3.7.21 GetVidTotalTimeMS
- 3.7.22 GetVidCaptionSyncOffset
- 3.7.23 GetVidCaptionSyncOffset

3.8 Record Mode Get Parameter FS Command

- 3.8.1 GetRecLineInbps
- 3.8.2 GetRecVoicebps
- 3.8.3 GetRecFMbps
- 3.8.4 GetRecLineInVolume
- 3.8.5 GetRecVoiceVolume
- 3.8.6 GetRecAutoSync
- 3.8.7 GetRecMode
- 3.8.8 GetRecCurTime
- 3.8.9 GetRecTotalTime
- 3.8.10 GetRecFileName

3.9 FM Radio Mode Get Parameter FS Command

- 3.9.1 GetFmrStereo
- 3.9.2 GetFmrRegion
- 3.9.3 GetFmrMinFreq
- 3.9.4 GetFmrMaxFreq
- 3.9.5 GetFmrCurrentFreq
- 3.9.6 GetFmrPresetOn
- 3.9.7 GetFmrPresetIndex
- 3.9.8 GetFmrNonePresetIndex
- 3.9.9 GetFmrPresetFreq
- 3.9.10 GetFmrNearPresetIndex
- 3.9.11 GetFmrPresetName
- 3.9.12 GetFmrScanState

3.10 Mobile TV(DMB/DAB) Mode Get Parameter FS Command(Domestic Version Only)

- 3.10.1 GetMTVAlarmState
- 3.10.2 GetMTVRecState
- 3.10.3 GetMTVRecRemainTime
- 3.10.4 GetMTVRecTime
- 3.10.5 GetMTVRegion
- 3.10.6 GetMTVSLS
- 3.10.7 GetMTVDLS
- 3.10.8 GetMTVSNR
- 3.10.9 GetMTVEnsLable
- 3.10.10 GetMTVSvcLable
- 3.10.11 GetMTVSamplerate
- 3.10.12 GetMTVBitrate
- 3.10.13 GetMTVSvcType
- 3.10.14 GetMTVTotalSvcldx

3.10.15 GetMTVScanStatus

3.10.16 GetMTVScanInfo

3.10.17 GetMTVDMBReady

3.10.18 GetMTVPreviewScan

3.11 Etc. Get Parameter FS Command

3.11.1 GetEtcUIConfig

3.11.2 GetEtcSettingString

3.11.3 GetEtcSettingCurrIndex

3.11.4 GetEtcSettingStep

3.11.5 GetEtcSettingTitle

3.11.6 GetEtcVolume

3.11.7 GetEtcTotalPLNum

3.11.8 GetEtcCurPLIndex

3.11.9 GetEtcFileName

3.11.10 GetEtcState

3.11.11 GetEtcFavorite

3.11.12 GetEtcTASValue

3.11.13 GetEtcOpenState

3.11.14 GetEtcTASValueX

3.11.15 GetEtcTASValueY

3.11.16 GetEtcTASValueZ

4 Set Parameter FS Command

4.1 JetEffect 2.0 Set Parameter FS Command

4.1.1 SetJetEffectIndex

4.1.2 SetJetUserIndex

4.1.3 SetJetUserBBE

4.1.4 SetJetUserM3B

4.1.5 SetJetUser3DS

4.1.6 SetJetUserMPE

4.1.7 SetJetUserSTE

4.1.8 SetJetUserEQBandGain

4.1.9 SetJetUserEQCenterFreq

4.1.10 SetJetUserEQBandWidth

4.1.11 SetJetSettingMode

4.2 Display Set Parameter FS Command

4.2.1 SetDisLanguage

4.2.2 SetDisFont

4.2.3 SetDisPlayTime

4.2.4 SetDisLyrics

4.2.5 SetDisLCDAutoOff

4.2.6 SetDisBrightness

4.2.7 SetDisTVOutMode

4.2.8 SetDisWallpaper

4.2.9 SetDisLCDAutoOffReset

4.3 Timer Set Parameter FS Command

4.3.1 SetTimTime

4.3.2 SetTimAlarmDuration

4.3.3 SetTimAlarmRepeat

4.3.4 SetTimAlarmMode

4.3.5 SetTimAlarmTime

4.3.6 SetTim24HDisplay

4.3.7 SetTimSleep

4.3.8 SetTimAutoOff

4.4 Bluetooth Set Parameter FS Command

4.4.1 SetBTHState

4.4.2 SetBTHStartScan

4.4.3 SetBTHStopScan

4.4.4 SetBTHConnect

4.4.5 SetBTHRemoveList

4.4.6 SetBTHPinCode

4.5 System Set Parameter FS Command

4.5.1 SetSysCtrlHoldState

4.5.2 SetSysPan

4.5.3 SetSysUSBMode

4.5.4 SetSysDefault

4.5.5 SetSysClearHDS

4.5.6 SetSysSleepMode

4.5.7 SetSysBeepMode

4.5.8 SetSysOffFlag

4.6 Music Mode Set Parameter FS Command

4.6.1 SetAudSkipLength

4.6.2 SetAudScanSpeed

4.6.3 SetAudResume

4.6.4 SetAudPSpeed

4.6.5 SetAudFadeln

4.6.6 SetAudRepeat

4.6.7 SetAudShuffle

4.6.8 SetAudBoundary

4.6.9 SetAudAlbumArtIndex

4.6.10 SetAudLDBConfig

4.6.11 SetAudPitchControl

4.7 Video Mode Set Parameter FS Command

4.7.1 SetVidReOpen

4.7.2 SetVidATime

4.7.3 SetVidBTime

4.7.4 SetVidABOff

4.7.5 SetVidSkipLength

4.7.6 SetVidResume

4.7.7 SetVidPSpeed

4.7.8 SetVidCaption

4.7.9 SetVidRepeat

4.7.10 SetVidBoundary

4.7.11 SetVidColorKey

4.7.12 SetVidAlpha

4.7.13 SetVidStartMode

- 4.7.14 SetVidSCR
- 4.7.15 SetVidCaptionSyncOffset
- 4.7.16 SetVidTVOut
- 4.7.17 SetVidUIEnable
- 4.7.18 SetVidCaptionEnable

4.8 Record Mode Set Parameter FS Command

- 4.8.1 SetRecLineInbps
- 4.8.2 SetRecVoicebps
- 4.8.3 SetRecFMbps
- 4.8.4 SetRecLineInVolume
- 4.8.5 SetRecVoiceVolume
- 4.8.6 SetRecAutoSync
- 4.8.7 SetRecMode

4.9 FM Radio Mode Set Parameter FS Command

- 4.9.1 SetFmrStereo
- 4.9.2 SetFmrAutoScan
- 4.9.3 SetFmrRegion
- 4.9.4 SetFmrCurrentFreq
- 4.9.5 SetFmrPresetOn
- 4.9.6 SetFmrPresetCurNum
- 4.9.7 SetFmrPresetFreq

4.10 Mobile TV(DMB/DAB) Mode Set Parameter FS Command(Domestic Version Only)

- 4.10.1 SetMTVFastPrevScan
- 4.10.2 SetMTVMakeDMBThumb
- 4.10.3 SetMTVDelDMBThumb
- 4.10.4 SetMTVRegion
- 4.10.5 SetMTVPreviewScan

4.11 Etc. Set Parameter FS Command

- 4.11.1 SetEtcUIConfig
- 4.11.2 SetEtcSettingInit
- 4.11.3 SetEtcSettingValue
- 4.11.4 SetEtcSettingNext
- 4.11.5 SetEtcSettingPrev
- 4.11.6 SetEtcCurPLIndex
- 4.11.7 SetEtcLCDTimerOff
- 4.11.8 SetEtcWallpaper

5 Etc. FS Command

5.1 Browser FS Command

- 5.1.1 EtcBrwSetInitialization
- 5.1.2 EtcBrwSetNextStage
- 5.1.3 EtcBrwSetPrevStage
- 5.1.4 EtcBrwGetListString
- 5.1.5 EtcBrwGetTitleString
- 5.1.6 EtcBrwGetCurIndex
- 5.1.7 EtcBrwGetFolderNum
- 5.1.8 EtcBrwAddFavorite

- 5.1.9 EtcBrwDelete
- 5.1.10 EtcBrwGetType

5.2 Mode FS Command

- 5.2.1 EtcModChangeMode
- 5.2.2 EtcModSetWidgetMode

5.3 Document FS Command

- 5.3.1 EtcTxtOpen
- 5.3.2 EtcTxtRead
- 5.3.3 EtcTxtNextPage
- 5.3.4 EtcTxtNextLine
- 5.3.5 EtcTxtPrevPage
- 5.3.6 EtcTxtPrevLine
- 5.3.7 EtcTxtGetTotalSize
- 5.3.8 EtcTxtGetCurOffset
- 5.3.9 EtcTxtGotoOffset
- 5.3.10 EtcTxtGetOnePageSize
- 5.3.11 EtcTxtSetBookmark
- 5.3.12 EtcTxtSetFontize
- 5.3.13 EtcTxtSetLineCount
- 5.3.14 EtcTxtSetDisplayWidth
- 5.3.15 EtcTxtSetRepeatCount
- 5.3.16 EtcTxtGetLastPageOffset

5.4 User Data FS Command

- 5.4.1 EtcUsrGetNumber
- 5.4.2 EtcUsrSetNumber
- 5.4.3 EtcUsrGetString
- 5.4.4 EtcUsrSetString
- 5.4.5 EtcUsrGetMainmenu
- 5.4.6 EtcUsrSetMainmenu

5.5 Notepad FS Command

- 5.5.1 EtcNpdSave
- 5.5.2 EtcNpdCheck
- 5.5.3 EtcNpdDelete
- 5.5.4 EtcNpdCaptureBuffer
- 5.5.5 EtcNpdCheckBuffer
- 5.5.6 EtcNpdClearBuffer

1 FS Command

1.1 FS Command Summary

FS Commands are interface functions created to control the 'Low Level' class in the ActionScript. The S9 separately defines 'ext_fscommand2()' to avoid conflicts with conventional FS Command functions. This function is used only in the Flash engine built in to the S9.

```
Example) var value:Number = ext_fscommand2("SetDisBrightness", "5");
```

FS Commands are controlled by predefined commands in the firmware and their input value. In the example above, 'SetDisBrightness' controls the brightness and the value '5' is the input value. If a false command such as 'SetDisBright' is entered or an FS Command is called with a value that is out of the input value range (0-5), the value of '-1' is saved as the value variable. Generally, if an FS Command is called in a form different than what is predefined in the firmware, the return value of '-1' is returned. Therefore, the ActionScript shows with the return value after the calling of the FS Command whether the process was successful or an error has occurred.

```
Example) var alarmTime:String;  
var value:Number = ext_fscommand2("GetTimAlarmTime", "alarmTime");
```

There is also an FS command that converts the data into a string. The example above uses an FS Command that gets the alarm time to save the alarm time that is currently defined in the 'alarmTime' variable in a "hours|minutes" format. However, take note that this does not work properly if the variable is declared as a local variable. Therefore, when processing a string with an FS Command, the variable must be declared as a global or timeline variable before it can be used.

1.2 Key FS Command

Explanation of Key FS Commands: Key FS Commands are defined for the purpose of controlling the 'Low Level Class' according to the user manipulation (touch, hardware keys, etc.).

1.3 Get/Set Parameter FS Command

Explanation on FS Commands that obtain or configure parameters used in the UI: Get Parameter FS Command has been created for the purpose of obtaining setting values and does not directly control the Low Level class. Set Parameter FS Command, however, includes those that only save setting values as well as those that directly control the Low Level class.

Also, Get/Set Parameter includes a mix of, largely, setting values that are controlled in the settings menu and setting values that are controlled directly. For example, the play speed ('GetAudPSpeed', 'SetAudPSpeed') currently available in the Music Mode Get Parameter FS Command is only available in the settings menu. However, depending on the needs of the UCI developer, it is also possible to control the play speed on the music screen by building in a UI on the music screen and then adding the relevant FS Command. Furthermore, while the playback boundary can only be controlled on the music screen, a UI can be added in the settings menu so that the playback boundary can be controlled in the settings menu.

Only, in order to be able to process various languages, the settings menu actually available on the S9 has been created with independent FS Commands in a tree structure similar to the browser. This will be further explained in the Etc Set Parameter FS Command section.

Finally, while there are FS Commands that can accommodate both the Get and Set actions, there are also commands such as 'GetAudTitle' which only processes the Get action or 'SetAudLDBConfig' which only processes the Set action.

1.4 Etc FS Command

In the Etc FS Command, there are FS Commands that are characteristically different from the Key, Get/Set Parameter commands. These control the browser, the mode, the document, and the user data.

2 Key FS Command

2.1 Common Mode Key FS Command

Explanation of Key FS Commands commonly used regardless of the mode: The basic UI design principle of the S9 is to control the volume when there is +, - input except under special circumstances such as the browser and the settings menu. Since this is merely a UI design principle of the S9, it is also possible to instigate actions other than volume control by setting the +, - key input to call different FS Commands. Furthermore, it is also possible to control the volume upon +, - key input in the browser and the settings menu

2.1.1 KeyComPlus

a) Explanation of Commands : Increases the volume by one step. This only works when the volume is below maximum (40). When the maximum value has been reached, only the maximum value is returned without changing the volume. Volume is controlled one step at a time in order to protect hearing.

b) Input Value : None

c) Return Value

- '-1' : When Volume isn't set because Recording function is activated

- '0 ~ 40' : Return Modified Volume Value when it's set properly

d) Example

```
var value:Number = ext_fscommand2("KeyComPlus");
```

2.1.2 KeyComMinus

a) Explanation of Commands : Decreases the volume by one step. This only works when the volume is above minimum (0). When the minimum value has been reached, only the minimum value is returned without changing the volume. Volume is controlled one step at a time in order to protect hearing.

b) Input Value : None

c) Return Value

- '-1' : When Volume isn't set because Recording function is activated

- '0 ~ 40' : Return Modified Volume Value when it's set properly

d) Example

```
var value:Number = ext_fscommand2("KeyComMinus");
```

2.2 Music Mode Key FS Command

Explain Key FS Command used in Music Mode

2.2.1 KeyAudPlay

a) Explanation of Commands: Play Music.

b) Input Value : None

c) Return Value

- '-1' : When it can't play due to the reason that there's no music files in Play list

- '1' : When it starts playing normally

d) Example

```
var value:Number = ext_fscommand2("KeyAudPlay");
```

2.2.2 KeyAudPause

a) Explanation of Commands: Stop playing music temporarily

b) Input Value : None

c) Return Value

- '-1' : When It can't activate temporary stop due to the reason that It is called while music stopped
- '1' : When normally music stop temporary

d) Example

```
var value:Number = ext_fscommand2("KeyAudPause");
```

2.2.3 KeyAudStop

a) Explanation of Commands : Stop Music playback

b) Input Value : None

c) Return Value

- '-1' : When It can't activate temporary stop due to the reason that It is called while music stopped
- '1' : When it is stopped normally

d) Example

```
var value:Number = ext_fscommand2("KeyAudStop");
```

2.2.4 KeyAudShortFF

a) Explanation of Commands : Skips (FF) during playback according to the skip interval

('GetAudSkipLength' FS Command). If the music not playing, it skips to the next song.

b) Input Value : None

c) Return Value

- '1' : When it works normally

d) Example

```
var value:Number = ext_fscommand2("KeyAudShortFF");
```

2.2.5 KeyAudShortREW

a) Explanation of Commands : Skips (REW) during playback according to the skip interval

('GetAudSkipLength' FS Command). If the music not playing, it skips to the previous song.

b) Input Value : None

c) Return Value

- '1' : When it works normally

d) Example

```
var value:Number = ext_fscommand2("KeyAudShortREW");
```

2.2.6 KeyAudLongFF

a) Explanation of Commands : Scans (FF) during playback according to the scan speed

('GetAudScanSpeed' FS Command). If the music not playing, it skips to the next song.

b) Input Value : None

c) Return Value

- '1' : When it works normally

d) Example

```
var value:Number = ext_fscommand2("KeyAudLongFF");
```

2.2.7 KeyAudLongREW

a) Explanation of Commands : Scans (REW) during playback according to the scan speed

('GetAudScanSpeed' FS Command). If the music not playing, it skips to the previous song.

- b) Input Value : None
- c) Return Value
 - '1' : When it works normally
- d) Example


```
var value:Number = ext_fscommand2("KeyAudLongREW");
```

2.2.8 KeyAudDirectSeek

- a) Explanation of Commands : Set the location of Playback with the second (unit)
- b) Input Value : Time to be shifted ((unit) Second)
- c) Return Value
 - '-1' : When the location of playback isn't controlled due to the reason that the location of playback is set more than the total playback time.
 - '1' : When it normally sets the location of playback
- d) Example


```
var value:Number = ext_fscommand2("KeyAudDirectSeek", "120");
```

2.2.9 KeyAudBookmark

- a) Explanation of Commands : Add current music files on the bookmark list
- b) Input Value : None
- c) Return Value
 - '-1' : When it can't be added due to the reason that the number of music bookmark list exceeds the limit
 - '0 ~ 255' : Return the added Index value when it is normally added on the bookmark list
- d) Example


```
var value:Number = ext_fscommand2("KeyAudBookmark");
```

2.2.10 KeyAudFavorite

- a) Explanation of Commands : Add the current Music file on the favorite music file list
- b) Input Value : None
- c) Return Value
 - '-1' : When it can't be added due to the reason that the number of favorite music file list exceeds the limit
 - '0 ~ 255' : Return the added Index value when it is normally added on the favorite music list
- d) Example


```
var value:Number = ext_fscommand2("KeyAudFavorite");
```

2.2.11 KeyAudABMode

- a) Explanation of Commands : Sets the repeat interval on the file that is currently being currently.
Every time the FS Command is executed, the action repeats in the order of 'Off -> A -> A-B -> Off'.
As a note, the 'GetAudABMode' FS Command can also show the status value of the repeat interval that is currently set.
- b) Input Value : None
- c) Return Value
 - '-1' : When it fails to add it on the list due to exceeding the maximum number of files
 - '0 ~ 255' : Return the added index value when it normally adds it on the favorite list
- d) Example


```
var value:Number = ext_fscommand2("KeyAudABMode");
```

2.3 Video Mode Key FS Command

Explaining Key FS Command used in Video Mode

2.3.1 KeyVidInit

- a) Action Explanation : Setting the environment which is necessary for Video
- b) Input Value : None
- c) Output Value
 - '-1' : When it doesn't work normally
 - '0' : When Video Environment is set normally
- d) Example

```
ext_fscommand2("KeyVidInit");
```

2.3.2 KeyVidDeinit

- a) Action Explanation : Canceling the environment for Video
- b) Input Value : None
- c) Output Value
 - '-1' : When it doesn't work normally
 - '0' : When it cancel the environment

```
ext_fscommand2("KeyVidDeinit");
```

2.3.3 KeyVidOpen

- a) Action Explanation : Open the file with the value of file list index
- b) Input Value : File list index
- c) Output Value
 - '-1' : When it doesn't work normally
 - '1' : When it succeeds in opening a file

```
/* Sample for KeyVidOpen,  
: Example for opening files */
```

```
var iCur, iLast, iTrg;
```

```
// get index  
iCur = ext_fscommand2("GetEtcCurPLIndex");  
iLast = ext_fscommand2("GetEtcTotalPLNum")-1;  
iTrg = iCur++;  
if(iTrg > iLast) iTrg = iLast;  
ext_fscommand2("KeyVidOpen", iTrg);
```

2.3.4 KeyVidPlay

- a) Action Explanation : Playing the currently selected file
- b) Input Value : None
- c) Output Value
 - '-1' : When it fails to work normally
 - '1' : When it works normally

```
ext_fscommand2("KeyVidPlay");
```

2.3.5 KeyVidPause

- a) Action Explanation : Temporarily stop the selected file
- b) Input Value : None
- c) Output Value

- '-1' : When it fails to work properly
- '1' : When it works normally

d) Example

```
ext_fscommand2("KeyVidPause");
```

2.3.6 KeyVidLongREW

a) Action Explanation : Move to the previous Key Frame

b) Input Value : None

c) Output Value

- '-1' : When the command fails to work normally
- '1' : When it moves to the previous key frame normally
- '0' : When there is no file

d) Example

```
ext_fscommand2("KeyVidLongREW ");
```

2.3.7 KeyVidLongFF

a) Action Explanation : Move to the next key frame

b) Input Value : None

c) Output Value

- '-1' : When the command fails to work normally
- '1' : When it moves to the next key frame normally
- '0' : When there is no file

d) Example

```
ext_fscommand2("KeyVidLongFF ");
```

2.3.8 KeyVidDirectSeek

a) Action Explanation : Move to the near key frame when you want to

b) Input Value : The wanted time, A unit is a millisecond

c) Output Value

- '-1' : When the command fails to work normally
- '1' : When it finds the key frame normally
- '0' : When there is no file

d) Example

```
/* Sample for KeyVidDirectSeek
```

```
 : Move to ten seconds */
```

```
var trgSec    = 10;
```

```
var totalSec  = ext_fscommand2("GetVidTotalTime");
```

```
if(trgSec > totalSec) trace("Seek error");
```

```
ext_fscommand2("KeyVidDirectSeek", trgSec*1000);
```

```
ext_fscommand2("KeyVidLongKeyRelease");
```

2.3.9 KeyVidLongKeyRelease

a) Action Explanation : Complete the task of KeyVidDirectSeek. WHEN KeyVidDirectSeek is being called, KeyVidLongKeyRelease shall be called to complete the task.

b) Input Value : None

c) Output Value

- '-1' : When the command fails to work normally
- '1' : When it's normal
- '0' : When there is no file

d) Example

Please refer to KeyVidDirectSeek Example

2.3.10 KeyVidBookmark

a) Action Explanation : Bookmark the currently watched location

b) Input Value : None

c) Output Value

- '-1' : When it fails to add due to that the number of Music Bookmark list exceeds the limit or etc
- '0~255' : Return the added index when it is normally added on Music Bookmark list

d) Example

```
var idx = ext_fscommand2("KeyVidBookmark");
trace(idx + " th bookmark is added properly.");
```

2.3.11 KeyVidFavorite

a) Action Explanation : Add the current file on Favorite

b) Input Value : None

c) Output Value

- '-1' : When it fails to add due to that the number of favorite video list exceeds the limit or etc
- '0~255' : Return the added index value when it is added on favorite video list

d) Example

```
var idx = ext_fscommand2("KeyVidFavorite ");
trace(idx + " it is added as the ?th favorite.");
```

2.3.12 KeyVidCapture

a) Action Explanation : Capture a current display, The captured file is saved in COWON S9\Recordings\Video Capture. The type of captured file is as follows <file name>-<hh:24>:<mm>:<ss>.jpg.
(h: Time, m: Minute, s: Second)

b) Input Value : None

c) Output Value

- '-1' : When the command fails to work normally
- '1' : When the capture works normally

d) Example

```
ext_fscommand2("KeyVidCapture ");
```

2.3.13 KeyVidMakeScene

a) Action Explanation : Create Video clip with the entered index

b) Input Value : Video Clip Index, The total number of indexes is 12(0 ~ 11).

c) Output Value

- '-1' : When the command fails to work normally
- '1' : When it is created

d) Example

```
/* Sample for KeyVidMakeScene
: Create all Video Clips. */
for(var i = 0; i < 12; i++)
```

```

{
    // Check if there is any video clip.
    if(ext_fscommand2("GetVidScene", i) == 0)
    {
        // When there is no video clip, it is created
        ext_fscommand2("KeyVidMakeScene",
i);
    }
}

// completing video clip creation
ext_fscommand2("KeyVidSceneInit");

```

2.3.14 KeyVidSceneInit

- a) Action Explanation : Completing Video Clip creation and initialization
- b) Input Value : None
- c) Output Value
 - '-1' : When the command fails to work normally
 - '1' : When it completes the video clip creation normally
- d) Example
 - Please refer to KeyVidMakeScene

2.4 FM Radio Mode Key FS Command

Explanation of Key FS Commands used in the FM Radio Mode: The radio operates differently in the Low Level class depending on the region. For example, the effective radio frequencies in 'Japan' are 76 MHz – 108 MHz, whereas 'Countries Other than Japan' is 87.5 MHz – 108 MHz. Also, the frequency change increment is 100 kHz (unit) for 'Korea', 'Japan', and 'USA', whereas it is 50 kHz (unit) for 'Europe and China'. This is because the FM radio system is different depending on the country.

2.4.1 KeyFmrRecStart

- a) Explanation of Commands: Start FM Radio Recording
- b) Input Value : None
- c) Return Value
 - '-1' : When it fails to record due to that there is not enough space
 - '1' : When it normally starts recording
- d) Example
 - var value:Number = ext_fscommand2("KeyFmrRecStart");

2.4.2 KeyFmrRecStop

- a) Explanation of Commands : Stop FM Radio Recording
- b) Input Value : None
- c) Return Value
 - '-1' : When it fails to stop recording due to that FS Command isn't entered in a state of FM Radio Recording
 - '1' : When it normally stops recording
- d) Example
 - var value:Number = ext_fscommand2("KeyFmrRecStop");

2.4.3 KeyFmrShortFF

- a) Explanation of Commands : Increases the radio frequency from the current frequency according to the

regional radio frequency change increment unit (100 kHz or 50 kHz). If the current frequency is the maximum frequency, the frequency is changed to each country's minimum frequency value. If a radio preset ('SetFmrPresetOn') is currently in operation, this moves the frequency to the next preset.

b) Input Value : None

c) Return Value

- '76000 ~ 118000' : Return the modified frequency value when it works properly
(Japan : 76000 ~ 108000, Other countries : 87500 ~ 108000)

d) Example

```
var value:Number = ext_fscommand2("KeyFmrShortFF");
```

2.4.4 KeyFmrShortREW

a) Explanation of Commands : Decreases the radio frequency from the current frequency according to the regional radio frequency change increment unit (100 kHz or 50 kHz). If the current frequency is the minimum frequency, the frequency is changed to each country's maximum frequency value. If a radio preset ('SetFmrPresetOn') is currently in operation, this moves the frequency to the previous preset.

b) Input Value : None

c) Return Value

- '76000 ~ 118000' : Return the modified frequency value when it works properly
(Japan : 76000 ~ 108000, Other countries : 87500 ~ 108000)

d) Example

```
var value:Number = ext_fscommand2("KeyFmrShortREW");
```

2.4.5 KeyFmrLongFF

a) Explanation of Commands : Scans and sets the frequency to the next closest broadcasting station. If a radio preset ('SetFmrPresetOn') is currently in operation, this moves the frequency to the next preset.

b) Input Value : None

c) Return Value

- '76000 ~ 118000' : Return the modified frequency value when it works properly
(Japan : 76000 ~ 108000, Other countries : 87500 ~ 108000)

d) Example

```
var value:Number = ext_fscommand2("KeyFmrLongFF");
```

2.4.6 KeyFmrLongREW

a) Explanation of Commands : Scans and sets the frequency to the previous closest broadcasting station. If a radio preset ('SetFmrPresetOn') is currently in operation, this moves the frequency to the previous preset.

b) Input Value : None

c) Return Value

- '76000 ~ 118000' : Return the modified frequency value when it works properly
(Japan : 76000 ~ 108000, Other countries : 87500 ~ 108000)

d) Example

```
var value:Number = ext_fscommand2("KeyFmrLongREW");
```

2.5 Mobile TV(DMB/DAB) Mode Key FS Command

Key FS Command used in Mobile TV Mode is going to be released

2.6 Record Mode Key FS Command

Explanation of Key FS Commands used in the Record Mode: The Record Mode allows the user to record through line-in or a microphone. For radio and DMB/DAB recording, FS Commands need to be defined

separately in the Radio or Mobile TV Mode.

Because recording requires much of the device's performance, actions such as Flash animation effects which require a heavy load may interfere with the recording. Also, during the recording, there may be limitation on certain FS Commands and functions such as Load Movie. Therefore, it is not recommended to use FS Commands and functions such as Load Movie other than those that stop the recording such as 'KeyRecStop'. The same goes for radio and DMB/DAV recordings.

2.6.1 KeyRecStart

a) Explanation of Commands : Start Recording

b) Input Value : None

c) Return Value

- '-1' : When it fails to record due to that there's not enough space

- '1' : When it works normally

d) Example

```
var value:Number = ext_fscommand2("KeyRecStart");
```

2.6.2 KeyRecStop

a) Explanation of Commands : Stop recording or playing recorded files

b) Input Value : None

c) Return Value

- '-1' : When it fails to stop due to that FS Command isn't entered in a state of recording or playing recorded files

- '1' : When it stops normally

d) Example

```
var value:Number = ext_fscommand2("KeyRecStop");
```

2.6.3 KeyRecPause

a) Explanation of Commands : Stop recording or playing recorded files temporarily

b) Input Value : None

c) Return Value

- '-1' : When it fails to stop temporarily due to that FS Command is not entered in a state of recording or playing recorded files

- '1' : When it stops temporarily

d) Example

```
var value:Number = ext_fscommand2("KeyRecPause");
```

2.6.4 KeyRecPlay

a) Explanation of Commands : Play the recent recorded files

b) Input Value : None

c) Return Value

- '-1' : When it fails to play the recent recorded files due to that there is no recorded files

- '1' : When it normally plays the recent recorded files

d) Example

```
var value:Number = ext_fscommand2("KeyRecPlay");
```

3 Get Parameter FS Command

3.1 JetEffect 2.0 Get Parameter FS Command

Explanation of FS Command that reads the set value of JetEffect 2.0

3.1.1 GetJetEffectIndex

a) Explanation of Commands : Return the JetEffect's index value that is currently set

b) Input Value : None

c) Return Value : '0 ~ 29'

- '0' : 'User 1'
- '1' : 'User 2'
- '2' : 'User 3'
- '3' : 'User 4'
- '4' : 'Normal'
- '5' : 'BBE'
- '6' : 'BBE VIVA'
- '7' : 'BBE VIVA 2'
- '8' : 'BBE Mach3Bass'
- '9' : 'BBE MP'
- '10' : 'BBE Headphone'
- '11' : 'BBE Headphone 2'
- '12' : 'BBE Headphone 3'
- '13' : 'Rock'
- '14' : 'Jazz'
- '15' : 'Classic'
- '16' : 'Ballad'
- '17' : 'Pop'
- '18' : 'Club'
- '19' : 'Funk'
- '20' : 'Hip Hop'
- '21' : 'Techno'
- '22' : 'Blues'
- '23' : 'Metal'
- '24' : 'Dance'
- '25' : 'Rap'
- '26' : 'Wide'
- '27' : 'X-Bass'
- '28' : 'Hall'
- '29' : 'Vocal'

d) Example

```
var value:Number = ext_fscommand2("GetJetEffectIndex");
```

3.1.2 GetJetUserIndex

a) Explanation of Commands : Returns the index value of the JetEffect 2.0 user that is currently set.

Since the JetEffect 2.0 user index is an FS Command used in controlling detailed setting values of user presets, the applicable preset value must be designated before controlling the user preset.

b) Input Value : None

c) Return Value : '0 ~ 3'

- '0' : 'User 1'

- '1' : 'User 2'
- '2' : 'User 3'
- '3' : 'User 4'

d) Example

```
var value:Number = ext_fscommand2("GetJetUserIndex");
```

3.1.3 GetJetUserData

a) Explanation of Commands : Return the current JetEffect 2.0's index value, and return JetEffect User Data to Variable as an array of strings

b) Input Value : None

c) Return Value : '0 ~ 3'

- '0' : 'User 1'
- '1' : 'User 2'
- '2' : 'User 3'
- '3' : 'User 4'

d) Example

```
var value:Number = ext_fscommand2("GetJetUserData", "Variable");
```

e) Example Explanation : If the string "1|2|3|0|4|10|0|11|1|1|12|2|13|3|0|14|0|1" is saved in the variable and the value variable return is 2, the following values are saved in the 'User 3' preset. For more detailed information, please refer to the BBE and EQ Explanation FS Command section.

- BBE = 1
- M3B = 2
- 3DS = 3
- MPE = Off
- STE = 4
- EQ1(-2dB, 80Hz, Narrow)
- EQ2(-1dB, 300Hz, Normal)
- EQ3(0dB, 1.1kHz, Wide)
- EQ4(1dB, 4.1kHz, Narrow)
- EQ5(2dB, 6.9kHz, Normal)

3.1.4 GetJetUserBBE

a) Explanation of Commands : Return the BBE value that is currently set ('SetJetUserIndex') in JetEffect 2.0 User

b) Input Value : None

c) Return Value : '0 ~ 10'

- '0' : BBE Off
- '1 ~ 10' : BBE Value

d) Example

```
var value:Number = ext_fscommand2("GetJetUserBBE");
```

3.1.5 GetJetUserM3B

a) Explanation of Commands : Return Mach3base Value that is saved in JetEffect 2.0 User as a current set ('SetJetUserIndex')

b) Input Value : None

c) Return Value : '0 ~ 10'

- '0' : Mach3Bass Off
- '1 ~ 10' : Mach3Bass Value

d) Example

```
var value:Number = ext_fscommand2("GetJetUserM3B");
```

3.1.6 GetJetUser3DS

- a) Explanation of Commands : Return 3D Surround Value that is saved in JetEffect 2.0 User as a current set ('SetJetUserIndex')
- b) Input Value : None
- c) Return Value : '0 ~ 10'
 - '0' : 3D Surround Off
 - '1 ~ 10' : 3D Surround Value
- d) Example

```
var value:Number = ext_fscommand2("GetJetUser3DS");
```

3.1.7 GetJetUserMPE

- a) Explanation of Commands : Return MP Enhance Value that is saved in JetEffect 2.0 User as a current set ('SetJetUserIndex')
- b) Input Value : None
- c) Return Value : '0 ~ 1'
 - '0' : MP Enhance Off
 - '1' : MP Enhance On
- d) Example

```
var value:Number = ext_fscommand2("GetJetUserMPE");
```

3.1.8 GetJetUserSTE

- a) Explanation of Commands : Return Stereo Enhance Value that is saved in JetEffect 2.0 User as a current set ('SetJetUserIndex')
- b) Input Value : None
- c) Return Value : '0 ~ 10'
 - '0' : Stereo Enhance Off
 - '1 ~ 10' : Stereo Enhance Value
- d) Example

```
var value:Number = ext_fscommand2("GetJetUserSTE");
```

3.1.9 GetJetUserEQBandGain

- a) Explanation of Commands : Return the 'N'st Band's EQ Gain Value that is saved in JetEffect 2.0 User as a current set ('SetJetUserIndex')
- b) Input Value : '0 ~ 4'
 - '0' : Band 1
 - '1' : Band 2
 - '2' : Band 3
 - '3' : Band 4
 - '4' : Band 5
- c) Return Value : '0 ~ 24'
 - '0' : -12 dB
 - '12' : 0 dB
 - '24' : 12 dB
- d) Example

```
var value:Number = ext_fscommand2("GetJetUserEQBandGain", "0");
```

3.1.10 GetJetUserEQCenterFreq

- a) Explanation of Commands : Return the 'N'st Band's EQ Center Frequency Value that is saved in

JetEffect 2.0 User as a current set ('SetJetUserIndex'). As Each Band has a different Center Frequency Value, it's necessary to configure it with 'C)' item during displaying GUI

b) Input Value : '0 ~ 4'

- '0' : Band 1
- '1' : Band 2
- '2' : Band 3
- '3' : Band 4
- '4' : Band 5

c) Return Value : '0 ~ 3'

- Band 1
 - '0' : 80 Hz
 - '1' : 105 Hz
 - '2' : 135 Hz
 - '3' : 175 Hz
- Band 2
 - '0' : 220 Hz
 - '1' : 300 Hz
 - '2' : 385 Hz
 - '3' : 500 Hz
- Band 3
 - '0' : 650 Hz
 - '1' : 780 Hz
 - '2' : 1.1 kHz
 - '3' : 1.4 kHz
- Band 4
 - '0' : 1.8 kHz
 - '1' : 2.4 kHz
 - '2' : 3.0 kHz
 - '3' : 4.1 kHz
- Band 5
 - '0' : 6.9 kHz
 - '1' : 9 kHz
 - '2' : 11.7 kHz
 - '3' : 13 Hz

d) Example

```
var value:Number = ext_fscommand2("GetJetUserEQCenterFreq", "1");
```

3.1.11 GetJetUserEQBandWidth

a) Explanation of Commands : Return the 'N'st Band's EQ Bandwidth Value that is saved in JetEffect 2.0 User as a current set ('SetJetUserIndex')

b) Input Value : '0 ~ 4'

- '0' : Band 1
- '1' : Band 2
- '2' : Band 3
- '3' : Band 4
- '4' : Band 5

c) Return Value : '0 ~ 2'

- '0' : Narrow
- '1' : Normal

- '2' : Wide

d) Example

```
var value:Number = ext_fscommand2("GetJetUserEQBandWidth", "2");
```

3.2 Display Get Parameter FS Command

The explanation of FS Command that brings the established Display Value

3.2.1 GetDisLanguage

a) Explanation of Commands : Return the currently established Language Value

b) Input Value : None

c) Return Value : '0 ~ 14'

- '0' : English

- '1' : Korea

- '2' : Czech

- '3' : Dutch

- '4' : French

- '5' : German

- '6' : Hungarian

- '7' : Italian

- '8' : Japanese

- '9' : Polish

- '10' : Russian

- '11' : Simplified Chinese

- '12' : Traditional Chinese

- '13' : Spanish

- '14' : Swedish

d) Example

```
var value:Number = ext_fscommand2("GetDisLanguage");
```

3.2.2 GetDisFont

a) Explanation of Commands : Return the currently established Font Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : System Font

- '1' : User Font

d) Example

```
var value:Number = ext_fscommand2("GetDisFont");
```

3.2.3 GetDisPlayTime

a) Explanation of Commands : Return the currently established Playback Time Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Played Time

- '1' : Remaining Time

d) Example

```
var value:Number = ext_fscommand2("GetDisPlayTime");
```

3.2.4 GetDisLyrics

a) Explanation of Commands : Return the currently established Lyric Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Lyrics Off
- '1' : Lyrics On

d) Example

```
var value:Number = ext_fscommand2("GetDisLyrics");
```

3.2.5 GetDisLCDAutoOff

a) Explanation of Commands : Return the currently established Automatic Display-off Value

b) Input Value : None

c) Return Value : '0 ~ 7'

- '0' : Off
- '1' : 5 seconds
- '2' : 10 seconds
- '3' : 20 seconds
- '4' : 30 seconds
- '5' : 40 seconds
- '6' : 50 seconds
- '7' : 60 seconds

d) Example

```
var value:Number = ext_fscommand2("GetDisLCDAutoOff");
```

3.2.6 GetDisBrightness

a) Explanation of Commands : Return the currently established Brightness Value

b) Input Value : None

c) Return Value : '0 ~ 5'

d) Example

```
var value:Number = ext_fscommand2("GetDisBrightness");
```

3.2.7 GetDisTVOutMode

a) Explanation of Commands : Return the currently established TV Standard Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : NTSC
- '1' : PAL

d) Example

```
var value:Number = ext_fscommand2("GetDisTVOutMode");
```

3.2.8 GetDisWallpaper

a) Explanation of Commands : Return the enable/disable input value for the current wallpaper.

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Normal background
- '1' : Wallpaper

d) Example

```
var value:Number = ext_fscommand2("GetDisWallpaper");
```

3.3 Timer Get Parameter FS Command

Explanation of the FS Command that reads the established value of Time, Alarm and Timer

3.3.1 GetTimAlarmDuration

- a) Explanation of Commands : Return the currently established Alarm Duration Value
- b) Input Value : None
- c) Return Value : '0 ~ 6'
 - '0' : Always on
 - '1' : 20 minutes
 - '2' : 40 minutes
 - '3' : 60 minutes
 - '4' : 80 minutes
 - '5' : 100 minutes
 - '6' : 120 minutes
- d) Example

```
var value:Number = ext_fscommand2("GetTimAlarmDuration");
```

3.3.2 GetTimAlarmRepeat

- a) Explanation of Commands : Return the currently established Alarm Repeat Value
- b) Input Value : None
- c) Return Value : '0 ~ 1'
 - '0' : Once
 - '1' : Daily
- d) Example

```
var value:Number = ext_fscommand2("GetTimAlarmRepeat");
```

3.3.3 GetTimAlarmMode

- a) Explanation of Commands : Return the currently established Alarm Wake up mode Value
- b) Input Value : None
- c) Return Value : '0 ~ 5'
 - '0' : Off
 - '1' : Music
 - '2' : FM Radio
 - '3' : FM Radio Recording
 - '4' : DMB/DAB(DMB Model Only)
 - '5' : DMB/DAB Recording(DMB Model Only)
- d) Example

```
var value:Number = ext_fscommand2("GetTimAlarmMode");
```

3.3.4 GetTimAlarmTime

- a) Explanation of Commands : Return the currently established Alarm Time.
Save at Variable as a type of "One minute"
- b) Input Value : String Type Variable Name
- c) Return Value : '1'
- d) Example

```
var alarmTime:String;  
var value:Number = ext_fscommand2("GetTimAlarmTime", "alarmTime");
```
- e) Example Explanation
If the current alarm time is set to 22 hours and 15 minutes, string "22|15" is saved in the alarmTime variable.
The range of hours is '0 hour' – '23 hours' and the range of minutes is '0 minute' – '59 minutes'.

3.3.5 GetTimAlarmState

a) Explanation of Commands : Returns the current alarm state value. The alarm action is processed in two circumstances as shown below. This FS Command, unlike other FS Commands, has the Read & clear attribute, which causes it to internally initialize to '0' (normal state) upon being called. Therefore, in the cases listed below, it is recommended to initiate the UI action for the alarm after reading this FS Command to initialize the internal value of the device.

- When the device turns on upon alarm occurrence: The Low Level class operates according to the value set in 'SetTimAlarmMode' and then read the 'EtcModGetResumeMode' FS Command in the Launcher class to load the necessary UI Flash contents. In this case, it does not need to be processed separately in the UCI.

- When the alarm is triggered with the device on: Triggers the 'F12' key.

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Normal State

- '1' : Alarm State

d) Example

```
var value:Number = ext_fscommand2("GetTimAlarmState");
```

3.3.6 GetTim24HDisplay

a) Explanation of Commands : Returns the time display value that is currently set. A GUI that displays time must first read the user-defined time display value through this FS Command to correctly display the GUI according to the value.

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Indicating 'AM / PM'

- '1' : 'Indicating '24 hours'

d) Example

```
var value:Number = ext_fscommand2("GetTim24HDisplay");
```

3.3.7 GetTimSleep

a) Explanation of Commands : Return the currently established Sleep Timer Value. This value initializes to '0', every time the device turns on.

b) Input Value : None

c) Return Value : '0 ~ 12'

- '0' : Off

- '1' : 10 minutes

- '2' : 20 minutes

- '3' : 30 minutes

- '4' : 40 minutes

- '5' : 50 minutes

- '6' : 60 minutes

- '7' : 70 minutes

- '8' : 80 minutes

- '9' : 90 minutes

- '10' : 100 minutes

- '11' : 110 minutes

- '12' : 120 minutes

d) Example

```
var value:Number = ext_fscommand2("GetTimSleep");
```

3.3.8 GetTimAutoOff

a) Explanation of Commands : Return the currently established Auto Off Time. If the cases are as follows, it works normally. Excepting following cases, the device turns off itself as the established value, unless you control the device.

- Music Play
- Video Play
- Listen to Radio
- Watch DMB or DAB
- Recording or Recorded File Play
- When Automatic Screen Timer is turned off by 'SetEtcLCDTimerOff' FS Command in Documents or Pictures, etc
- When users touch or input key (Initializing Auto Off Time based on the last input time)

b) Input Value : None

c) Return Value : '0 ~ 8'

- '0' : Off
- '1' : 1 minute
- '2' : 2 minutes
- '3' : 3 minutes
- '4' : 4 minutes
- '5' : 5 minutes
- '6' : 10 minutes
- '7' : 20 minutes
- '8' : 30 minutes

d) Example

```
var value:Number = ext_fscommand2("GetTimAutoOff");
```

3.4 Bluetooth Get Parameter FS Command

Explanation of the FS Command that reads the established Bluetooth Value

3.4.1 GetBTHState

a) Explanation of Commands: Return the currently established Bluetooth Value

b) Input Value : None

c) Return Value : '-1 ~ 1'

- '-1' : Off
- '0' : Initializing
- '1' : On

d) Example

```
var value:Number = ext_fscommand2("GetBTHState");
```

3.4.2 GetBTHConnect

a) Explanation of Commands : Return the current Bluetooth Accessing Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Accessing
- '1' : Accessed

d) Example

```
var value:Number = ext_fscommand2("GetBTHConnect");
```

3.4.3 GetBTHTScan

- a) Explanation of Commands : Return the current Bluetooth Scanning Value
- b) Input Value : None
- c) Return Value : '0 ~ 1'
 - '0' : No Search
 - '1' : Searching
- d) Example

```
var value:Number = ext_fscommand2("GetBTHTScan");
```

3.4.4 GetBTHTDeviceList

- a) Explanation of Commands : Return the registered Bluetooth devices' name. And it's saved in Variable as a type of "Devices'name|The status of connection|Devices'name|The status of connection...."
- b) Input Value : String Type Variable Name
- c) Return Value : '0 ~ 7'
 - '0 ~ 7' : Return the number of registered Bluetooth devices
- d) Example

```
var bthList:String;  
var value:Number = ext_fscommand2("GetBTHTState", "bthList");
```
- e) Example Explanation :

If there are two Bluetooth devices, 'AA' and 'BB', registered on the device and only 'BB' is connected, value 2 is saved in the value variable, and "AA|0|BB|1" is saved in the bthList variable. If there are no Bluetooth devices registered on the device, value 0 is saved in the value variable, and "" (Null) is saved in the bthList variable.

3.5 System Get Parameter FS Command

The explanation of the FS Command that reads System & State Value

3.5.1 GetSysCtrlHoldState

- a) Explanation of Commands : Return the currently established the Hold Mode Value
- b) Input Value : None
- c) Return Value : '0 ~ 1'
 - '0' : Hold Entire
 - '1' : Hold Touch Screen only
- d) Example

```
var value:Number = ext_fscommand2("GetSysCtrlHoldState");
```

3.5.2 GetSysPan

- a) Explanation of Commands : Return the currently established Audio L/R Value
- b) Input Value : None
- c) Return Value : '0 ~ 20'
 - '0 ~ 9' : Reduce the output of the right earphone. In the case of setting '0', the output of the right earphone is minimized.
 - '10' : The same output of right/left
 - '11 ~ 20' : Reduce the output of the left earphone, In the case of setting '20', the output of the left earphone is maximized.
- d) Example

```
var value:Number = ext_fscommand2("GetSysPan");
```

3.5.3 GetSysUSBMode

- a) Explanation of Commands : Return the currently established USB Mode Value.
This FS Command always returns '0' with domestic firm wares
- b) Input Value : None
- c) Return Value : '0 ~ 1'
 - '0' : MSC
 - '1' : MTP
- d) Example

```
var value:Number = ext_fscommand2("GetSysUSBMode");
```

3.5.4 GetSysHoldKey

- a) Explanation of Commands : Return the current Hardware Hold Key Value
- b) Input Value : None
- c) Return Value : '0 ~ 1'
 - '0' : Off
 - '1' : On
- d) Example

```
var value:Number = ext_fscommand2("GetSysHoldKey");
```

3.5.5 GetSysBattery

- a) Explanation of Commands : Return the current Battery Value
- b) Input Value : None
- c) Return Value : '0 ~ 5'
 - '0 ~ 4' : Battery Min(0) ~ Battery Max(4)
 - '5' : Connecting the charger
- d) Example

```
var value:Number = ext_fscommand2("GetSysBattery");
```

3.5.6 GetSysSleepMode

- a) Explanation of Commands : Return the currently established Sleep Mode Value
- b) Input Value : None
- c) Return Value : '0 ~ 1'
 - '0' : Turning off leads to Power Off state
 - '1' : Turning off leads to Sleep state
- d) Example

```
var value:Number = ext_fscommand2("GetSysSleepMode");
```

3.5.7 GetSysBeepMode

- a) Explanation of Commands : Return the currently established Beep Value. Beep Sound occurs once when
Touch is punched
- b) Input Value : None
- c) Return Value : '0 ~ 4'
 - '0' : Off
 - '1' : Water drop
 - '2' : Balloon
 - '3' : Bell 1
 - '4' : Bell 2
- d) Example

```
var value:Number = ext_fscommand2("GetSysBeepMode");
```

3.5.8 GetSysTotalSize

- a) Explanation of Commands : Return the total space of the device
- b) Input Value : None
- c) Return Value : '0 ~'
 - '0 ~' : (unit) MB
- d) Example

```
var value:Number = ext_fscommand2("GetSysTotalSize");
```

3.5.9 GetSysFreeSize

- a) Explanation of Commands : Return the current free space
- b) Input Value : None
- c) Return Value : '0 ~'
 - '0 ~' : (unit) MB
- d) Example

```
var value:Number = ext_fscommand2("GetSysFreeSize");
```

3.5.10 GetSysVersion

- a) Explanation of Commands : Save the current Firmware Version in variable
- b) Input Value : String Type Variable Name
- c) Return Value : '1'
- d) Example

```
var version:String;  
var value:Number = ext_fscommand2("GetSysVersion", "version");
```
- e) Example Explanation :

If the firmware version is "1.23b", "1.23" is saved in the version variable and "1" is saved in the value variable.

3.6 Music Mode Get Parameter FS Command

The explanation of FS Command that reads Music Mode & State Value

3.6.1 GetAudSkipLength

- a) Explanation of Commands : Return the currently established Skip Interval Value (FF/REW Short Key)
- b) Input Value : None
- c) Return Value : '0 ~ 7'
 - '0' : Track
 - '1' : 2 Sec
 - '2' : 3 Sec
 - '3' : 4 Sec
 - '4' : 5 Sec
 - '5' : 10 Sec
 - '6' : 20 Sec
 - '7' : 30 Sec
- d) Example

```
var value:Number = ext_fscommand2("GetAudSkipLength");
```

3.6.2 GetAudScanSpeed

- a) Explanation of Commands : Return the currently established Seek Interval Value (FF/REW Long Key)
- b) Input Value : None

c) Return Value : '0 ~ 4'

- '0' : 3 Sec
- '1' : 5 Sec
- '2' : 10 Sec
- '3' : 20 Sec
- '4' : 30 Sec

d) Example

```
var value:Number = ext_fscommand2("GetAudScanSpeed");
```

3.6.3 GetAudResume

a) Explanation of Commands : Return the currently established Resume Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Off
- '1' : On

d) Example

```
var value:Number = ext_fscommand2("GetAudResume");
```

3.6.4 GetAudPSpeed

a) Explanation of Commands : Return the currently established Play Speed Value

b) Input Value : None

c) Return Value : '0 ~ 10'

- '0' : 50%
- '1' : 60%
- '2' : 70%
- '3' : 80%
- '4' : 90%
- '5' : 100%
- '6' : 110%
- '7' : 120%
- '8' : 130%
- '9' : 140%
- '10' : 150%

d) Example

```
var value:Number = ext_fscommand2("GetAudPSpeed");
```

3.6.5 GetAudFadeIn

a) Explanation of Commands : Return the currently established Fade In Value

b) Input Value : None

c) Return Value : '0 ~ 5'

- '0' : Off
- '1' : 1 sec
- '2' : 2 sec
- '3' : 3 sec
- '4' : 4 sec
- '5' : 5 sec

d) Example

```
var value:Number = ext_fscommand2("GetAudFadeIn");
```

3.6.6 GetAudRepeat

a) Explanation of Commands : Return the currently established Repeat Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Off

- '1' : On

d) Example

```
var value:Number = ext_fscommand2("GetAudRepeat");
```

3.6.7 GetAudShuffle

a) Explanation of Commands : Return the currently established Shuffle Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Off

- '1' : On

d) Example

```
var value:Number = ext_fscommand2("GetAudShuffle");
```

3.6.8 GetAudBoundary

a) Explanation of Commands : Return the currently established Boundary Value

b) Input Value : None

c) Return Value : '0 ~ 2'

- '0' : All files

- '1' : One song

- '2' : Folder

d) Example

```
var value:Number = ext_fscommand2("GetAudBoundary");
```

3.6.9 GetAudABMode

a) Explanation of Commands : Return the currently established AB Mode Value. This Value initializes itself to '0' every time the device turns on. Also, it is initialized when the music file which is being played finishes.

b) Input Value : None

c) Return Value : '0 ~ 2'

- '0' : Off

- '1' : A

- '2' : A-B

d) Example

```
var value:Number = ext_fscommand2("GetAudABMode");
```

3.6.10 GetAudBitRate

a) Explanation of Commands : Return the Bit Rate Value of the music file which is being played. It returns the Value normally only when the music file is being played or is stopped temporarily. But other than those cases, it returns '0'

b) Input Value : None

c) Return Value

- '0 ~' : (unit) kB

d) Example

```
var value:Number = ext_fscommand2("GetAudBitRate");
```

3.6.11 GetAudSampleRate

a) Explanation of Commands: Return the Sampling Rate Value of the music file which is being played.
It returns the Value normally only when the music file is being played or is stopped temporarily. But other than those cases, it returns '0'

b) Input Value : None

c) Return Value

- '0 ~' : (unit) Hz

d) Example

```
var value:Number = ext_fscommand2("GetAudSampleRate");
```

3.6.12 GetAudCodec

a) Explanation of Commands : Return the current music file type after saving it in Variable. It saves normally and returns '1' only when the music file is being played or is stopped temporarily. But other than those cases, it saves 'None' in Variable and returns '-1'

b) Input Value : String Type Variable Name

c) Return Value : '-1, '1'

d) Example

```
var codecType:String;  
var value:Number = ext_fscommand2("GetAudCodec", "codecType");
```

e) Example Explanation

If the type of the music file that is being played is the MP3 file, "MP3" is saved in Codectyoe Variable and its value is '1'

3.6.13 GetAudPlayTime

a) Explanation of Commands : Return the Play Time Value. It returns the Value normally only when the music file is being played or is stopped temporarily. But other than those cases, it returns '0'.

Yet, if 'GetDisPlayTime' FS Command Value is 'Remaining Time'('1'), the Play Time Value is calculated as follows: 'Total Play Time' – 'Played Time'

b) Input Value : None

c) Return Value

- '0 ~' : Sec (unit)

d) Example

```
var value:Number = ext_fscommand2("GetAudPlayTime");
```

3.6.14 GetAudTotalTime

a) Explanation of Commands : Return the Total Time Value. It returns the Value normally only when the music file is being played or is stopped temporarily. But other than those cases, it returns '0'.

b) Input Value : None

c) Return Value

- '0 ~' : (unit) a second

d) Example

```
var value:Number = ext_fscommand2("GetAudTotalTime");
```

3.6.15 GetAudAlbum

a) Explanation of Commands : This command saves to the variable the album name (tag) of the music file that belongs to the index input in the current playlist and returns the index input. If it cannot be processed normally, the command saves the 'unknown' string to the variable and returns '-1'.

b) Input Value : Index Value ('0' ~ ' The total number of play lists - 1'), String Type Variable Name

c) Return Value '-1 ~'

- '-1' : When the punched Index Value exceeds the limit or there's no Album Name in the file tag

- '0 ~' : Return the punched Index Value when it works normally

d) Example

```
var albumName:String;
```

```
var value:Number = ext_fscommand2("GetAudAlbum", "1", "albumName");
```

e) Example Explanation

If there are ten songs in the current playlist ('GetEtcTotalPLNum'), the valid index values are '0' – '9'.

If the file that belongs to index '1' has the name 'iAUDIO's Theme', the 'iAUDIO's Theme' string is saved to the variable and '1' is saved to the 'value' variable. The maximum length of a string is 255 characters.

3.6.16 GetAudArtist

This command saves to the variable the artist name (tag) of the music file that belongs to the index input in the current playlist and returns the index input. If it cannot be processed normally, the command saves the 'unknown' string to the variable and returns '-1'.

b) Input Value : Index Value ('0' ~ ' The total number of play lists - 1'), String Type Variable Name

c) Return Value '-1 ~'

- '-1' : When the punched Index Value exceeds the limit or there's no Singer's name in the file tag

- '0 ~' : Return the punched Index Value when it works normally

d) Example

```
var artistName:String;
```

```
var value:Number = ext_fscommand2("GetAudArtist", "1", "artistName");
```

e) Example Explanation

If there are ten songs in the current playlist ('GetEtcTotalPLNum'), the valid index values are '0' – '9'.

If the file that belongs to index '1' has the artist name 'Jack Lee', the 'Jack Lee' string is saved to the 'artistName' variable and '1' is saved to the 'value' variable. The maximum length of a string is 255 characters.

3.6.17 GetAudTitle

This command saves to the variable the song name (tag) of the music file that belongs to the index input in the current playlist and returns the index input. If it cannot be processed normally, the command saves the 'unknown' string to the variable and returns '-1'.

b) Input Value : Index Value ('0' ~ ' The total number of play lists - 1'), String Type Variable Name

c) Return Value '-1 ~'

- '-1' : When the punched Index Value exceeds the limit or there's no Song title in the file tag

- '0 ~' : Return the punched Index Value when it works normally

d) Example

```
var titleName:String;
```

```
var value:Number = ext_fscommand2("GetAudTitle", "1", "titleName");
```

e) Example Explanation

If there are ten songs in the current playlist ('GetEtcTotalPLNum'), the valid index values are '0' – '9'.

If the file that belongs to index '1' has the song name 'iAUDIO – Friends & Lovers', the 'iAUDIO – Friends & Lovers' string is saved to the 'titleName' variable and '1' is saved to the 'value' variable. The maximum length of a string is 255 characters.

3.6.18 GetAudAlbumArtTotalNum

This command returns the number of album artworks saved in the music file that belongs to the applicable

index input in the current playlist. Certain sound sources can save multiple album artworks to a single music file. The S9 supports up to six album artworks per song.

b) Input Value : Index Value ('0' ~ 'The total number of play lists - 1')

c) Return Value '-1', '1 ~ 6'

- '-1' : When the punched Index Value exceeds the limit or there's no Album Art in the file

- '1 ~ 6' : Return the number of Album Arts when it works normally

d) Example

```
var value:Number = ext_fscommand2("GetAudAlbumArtTotalNum", "1");
```

3.6.19 GetAudLDBType

a) Explanation of Commands : Return LDB(Lyric) Information of the music file which is being played.

LDB Information can be returned only when the music file is being played or is stopped temporarily.

b) Input Value : None

c) Return Value '0 ~ 2'

- '0' : NO LDB

- '1' : LDB 1.0

- '2' : LDB 2.0

d) Example

```
var value:Number = ext_fscommand2("GetAudLDBType");
```

3.6.20 GetAudPitchControl

a) Explanation of Commands: Return the pitch controlling value which is used for the playback speed except 100%.

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Off

- '1' : On

d) Example

```
var value:Number = ext_fscommand2("GetAudPitchControl");
```

3.7 Video Mode Get Parameter FS Command

Explains FS Command regarding Video setting and status variable.

3.7.1 GetVidWidth

a) Explanation of Commands : Calls for the Video file's width.

b) Input Value : None

c) Return Value

- '-1' : When not processed normally

- '0' : When the video file is corrupt.

- 'width': Video's width in pixel

d) Example

```
/* Sample for GetVidWidth
```

```
: Calls for video file's information. */
```

```
var videoWidth, videoHeight, videoFps;
```

```
var audioSampleRate, audioBitrate;
```

```
var fileName:String;
```

```
var curldx;
```

```

curlIdx = ext_fscommand2("GetEtcCurPLIndex");
ext_fscommand2("GetEtcFileName", curlIdx, "fileName");

videoWidth  = ext_fscommand2("GetVidWidth");
videoHeight = ext_fscommand2("videoHeight");
videoFps     = ext_fscommand2("videoFps");
audioSampleRate = ext_fscommand2("GetVidSampleRate");
audioBitrate = ext_fscommand2("GetVidBitRate");

trace("-----");
trace(" " + fileName);
trace("-----");
trace("-----");
trace(" Video Info");
trace("-----");
trace("Width Resolution : " + videoWidth);
trace("Height Resolution: " + videoWidth);
trace("Frame Rate: " + videoFps);
trace("-----");
trace("Audio Info");
trace("-----");
trace("Sample Rate: " + audioSampleRate);
trace("Beat Rate: " + audioBitrate);

```

3.7.2 GetVidHeight

- a) Explanation of Commands : Calls for video file's height.
- b) Input Value : None
- c) Return Value
 - '-1' : When not processed normally
 - '0' : When the video file is corrupt..
 - 'height': Video's height in pixel
- d) Example
 - Refer to GetVidWidth sample.

3.7.3 GetVidFrameRate

- a) Explanation of Commands : Calls for frame rate of the video file.
- b) Input Value : None
- c) Return Value
 - '-1' : When not processed normally
 - '0' : When the video file is corrupt..
 - 'framerate': frame/sec
- d) Example
 - Refer to GetVidWidth sample.

3.7.4 GetVidSampleRate

- a) Explanation of Commands : Calls for Audio sample rate of the video file.
- b) Input Value : None

- c) Return Value
 - '-1' : When not processed normally
 - '0' : When the video file is corrupt..
 - 'samplerate': Sample rate in Hz
- d) Example

Refer to GetVidWidth sample.

3.7.5 GetVidBitRate

- a) Explanation of Commands : Calls for audio's bitrate of the video file.
- b) Input Value : None
- c) Return Value
 - '-1' : When not processed normally
 - '0' : When the video file is corrupt..
 - 'bitrate': Bitrate in bit/sec.
- d) Example

Refer to GetVidWidth sample.

3.7.6 GetVidState

- a) Explanation of Commands : Calls for playback status of the video file.
- b) Input Value : None
- c) Return Value
 - '-1' : When not processed normally
 - '0' : Stopped.
 - '1': Playing.
- d) Example


```
/* Sample for GetVidState
: Play/Pause toggle button */
```

```
var playState = ext_fscommand2("GetVidState);
```

```
if(playState == 0){
    trace("it is stopped. Will play.");
    ext_fscommand2("KeyVidPlay");
}
else if(playState == 1){

    trace("It is playing. Will stop.");
    ext_fscommand2("KeyVidPause");
}
```

3.7.7 GetVidTVOutEnable

- a) Explanation of Commands : Calls for check on TV-Out cable connection.
- b) Input Value : None
- c) Return Value
 - '-1' : When not processed normally
 - '0' : Not connected
 - '1': Connected
- d) Example


```
/* Sample for GetVidTVOutEnable
```

: Sets for auto TV-Out when the cable is connected. Example */

```
TVOutObject.onEnterFrame = function()
{
    if(ext_fscommand2("GetVidTVOutEnable"))
    {
        trace("TV-out is connected, and will turn into TV-out mode");
        ext_fscommand2("SetVidTVOut", 1);
        delete this.onEnterFrame;
    }
}
```

3.7.8 GetVidStartMode

a) Explanation of Commands : Calls for 'start mode' of a video file. There are three different 'start mode' for video player.

1. 'Standard' (plays from the starting point)
2. 'Resume' (plays from the point left off from the last playback)
3. 'Bookmark/Favorite' (plays from bookmarked/favorite point)

b) Input Value : None

c) Return Value

- '-1' : When not processed normally
- '0' : 'Standard Mode'
- '1' : 'Resume Mode'
- '2' : 'Bookmark/favorite mode'

d) Example

```
/* Sample for GetVidStartMode
OpenMovieFile = function(idx:Number){

var mode          = ext_fscommand2("GetVidStartMode");
var resume        = ext_fscommand2("GetVidResume");

// Check to see if the file has been opened by bookmark/favorite from the outside of the video player.
if(mode == 2){
    // The file will be opened again, beginning from the point of bookmark/favorite.
    ext_fscommand2("SetVidReOpen");
}else{
    // Checks if resume mode is on, and if on, will change the starting mode.
    if(resume) ext_fscommand2("SetVidStartMode", 1);
}

// Opens the file.
ext_fscommand2("KeyVidOpen", idx);
}
```

3.7.9 GetVidOpenState

a) Explanation of Commands : Function that calls the result of file opening. F12 key event. Check according to F12 key event(refer to S9_Flash_guide_1.0.pdf 7Page).

b) Input Value : None

c) Return Value

- '-1' : When not processed normally
- '0' : Opening file has been normally processed.
- '1' : Opening file has failed.
- '2' : Stopped because play mode is not 'repeat'.
- '3' : Scheduled.
- '4' : TV-Out automatically ended. During TV-Out, device will return to normal mode after playing a file.

d) Example

```
/* Sample for GetVidOpenState
: Routine example that is processed according to F12 key event. */
TestKeyListener = function()
{
    if(Key.getCode() == 19) // Check F12 key
    {
        var result = ext_fscommand2("GetVidOpenState");
        switch(result)
        {
            case 0:
                trace("Successful opening the file.");
                break;

            case 1:
                trace("Failed to open the file.");
                break;

            case 2:
                trace("Video playback has ended.");
                break;

            case 4:
                trace("TV-out has ended.");
                break;

            case 3:
            default:
                break;
        }
    }
}
```

3.7.10 GetVidScene

a) Explanation of Commands : Check if a video clip exists in input index.

b) Input Value : Video clip index. Index consists of total 12 (0 ~ 11).

c) Return Value

- '-1' : When not processed normally
- '0' : Index does not exist.
- '1' : Index exist.

d) Example

Refer to example of KeyVidMakeScene.

3.7.11 GetVidSceneTime

a) Explanation of Commands : Brings the Video clip(index)'s time in mili seconds.

b) Input Value : Video clip index. Index consists of total 12 (0 ~ 11).

c) Return Value

- '-1' : When not processed normally
- 'video clip time' : In milliseconds

d) Example

```
/* Sample for GetVidSceneTime
: Brings the time of the video clip */
```

```
for(var i = 0; i < 12; i++){

    // Checks to see if video clip exist.
    if(ext_fscommand2("GetVidScene", i) == 0){
        var time = ext_fscommand2("GetVidSceneTime");
        trace("[ " + i + " ] " + "Time for video clip " + time + " milliseconds.");
    } else {
        // Video clip does not exist.
        trace("[ " + i + " ] Video clip does not exist."
    }
}
```

3.7.12 GetVidSkipLength

a) Explanation of Commands : Brings the skip length set from SETTINGS ->VIDEO / (VIDEO&DMB).

b) Input Value : None

c) Return Value

- '-1' : When not processed normally
- '0~7': Number of frames to be skipped.

d) Example

```
/* Sample for GetVidSkipLength
: KeyVidLongREW is controlled depending on the skip length set.*/
```

```
skipLength = ext_fscommand2("GetVidSkipLength")+1;
for(var i = 0; i < skipLength; i++)
    ext_fscommand2("KeyVidLongREW");
```

3.7.13 GetVidResume

a) Explanation of Commands : Brings the Resume value from SETTINGS ->VIDEO / (VIDEO&DMB).

b) Input Value : None

c) Return Value

- '-1' : When not processed normally
- '0': Resume deactivated
- '1': Resume activated.

d) Example

Refer to example of GetVidStartMode.

3.7.14 GetVidPSpeed

a) Explanation of Commands : Brings the playback speed value from SETTINGS ->VIDEO / (VIDEO&DMB).

b) Input Value : None

c) Return Value

- '-1' : When not processed normally

- '0~10' : Indicates that current playback speed is 50%~150% of the original speed.

d) Example

```
/* Sample for GetVidPSpeed
 : Routine for increasing playback speed by 10%. */
// Increase by 10%.
IncreasePlaySpeed = function()
{
    var curSpeedIdx = ext_fscommand2("GetVidPSpeed");
    trace("Current speed " + (curSpeedIdx*10) + 50 + " %");
    ext_fscommand2("SetVidPSpeed", (curSpeedIdx+1));
}
// Decrease by 10%.
DecreasePlaySpeed = function()
{
    var curSpeedIdx = ext_fscommand2("GetVidPSpeed");
    trace("Current speed " + (curSpeedIdx*10) + 50 + " %");
    ext_fscommand2("SetVidPSpeed", (curSpeedIdx-1));
}
```

3.7.15 GetVidCaption

a) Explanation of Commands : Brings the Subtitle value from SETTINGS ->VIDEO / (VIDEO&DMB).

b) Input Value : None

c) Return Value

- '-1' : When not processed normally

- '0' : Subtitle deactivated.

- '1' : Subtitle activated.

d) Example

/* Sample for GetVidCaption

: Shows the activation/deactivation status of subtitle by on/off setting. */

```
if(ext_fscommand2("GetVidCaption"))
ext_fscommand2("SetVidCaptionEnable, 1");
else
ext_fscommand2("SetVidCaptionEnable, 0");
```

3.7.16 GetVidRepeat

a) Explanation of Commands : Repeats the currently set repeat playback value.

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Off

- '1' : On

d) Example

```
var value:Number = ext_fscommand2("GetVidRepeat");
```

3.7.17 GetVidABMode

a) Explanation of Commands : Returns the currently set A-B mode status.

b) Input Value : None

c) Return Value :

- '0' : A-B off

- '1' : A-B on

d) Example

```
var VideoABModeState = ext_fscommand2("GetVidABMode");
```

3.7.18 GetVidPlayTime

a) Explanation of Commands : Brings the current video clip playback time in seconds.

b) Input Value : None

c) Return Value :

- '-1' : When not processed normally

- 'playtime' : Brings the current video clip playback time in seconds.

d) Example

```
/* Sample for GetVidPlayTime
```

```
: Displays the play time. */
```

```
PlayTimeObject.onEnterFrame = function()
```

```
{
```

```
    if(getTimer()-this.tick > 1000)
```

```
    {
```

```
        var msCur, msTot;
```

```
        var sCur, sTot;
```

```
        sCur = ext_fscommand2("GetVidPlayTime");
```

```
        sTot = ext_fscommand2("GetVidTotalTime");
```

```
        trace("play information in seconds: " + sCur + "/" + sTot);
```

```
        msCur = ext_fscommand2("GetVidPlayTimeMS");
```

```
        msTot = ext_fscommand2("GetVidTotalTimeMS");
```

```
        trace("play information in milliseconds: " + sCur + "/" + sTot);
```

```
        this.tick =getTimer();
```

```
    }
```

```
}
```

3.7.19 GetVidPlayTimeMS

a) Explanation of Commands : Brings the current video clip's play time in milliseconds.

b) Input Value : None

- c) Return Value :
 - '-1' : When not processed normally
 - 'playtime' : Brings the current video clip's play time in milliseconds.
- d) Example
 - Refer to example of GetVidPlayTime.

3.7.20 GetVidTotalTime

- a) Explanation of Commands : Brings the current video clip's entire play time in seconds.
- b) Input Value : None
- c) Return Value :
 - '-1' : When not processed normally
 - 'playtime' : Brings the current video clip's entire play time in seconds.
- d) Example
 - Refer to example GetVidPlayTime.

3.7.21 GetVidTotalTimeMS

- a) Explanation of Commands : Brings the current video file's entire play time in milliseconds.
- b) Input Value : None
- c) Return Value :
 - '-1' : When not processed normally
 - 'playtime' : Brings the current video file's entire play time in milliseconds.
- d) Example
 - Refer to example GetVidPlayTime.

3.7.22 GetVidCaptionSyncOffset

- a) Explanation of Commands : Brings the Subtitle synch offset in milliseconds.
- b) Input Value : None
- c) Return Value :
 - '-1' : When not processed normally.
 - "sync offset" : Brings the Subtitle synch offset in milliseconds.
- d) Example

```
/* Sample for GetVidCaptionSyncOffset
: displays the subtitle synch information. */
```

```
var syncOffset = ext_fscommand2("GetVidCaptionSyncOffset");
if(syncOffset > 0)
  trace("Current subtitle " + syncOffset + " is slow in milliseconds.");
else
  trace("Current subtitle " + syncOffset + " is fast in milliseconds.");
```

3.8 Record Mode Get Parameter FS Command

Explanation of FS Command that reads Recording State Value

3.8.1 GetRecLineInbps

- a) Explanation of Commands : Return the currently established Lin-in Quality Value.
- b) Input Value : None
- c) Return Value : '0 ~ 5'
 - '0' : 32 kbps
 - '1' : 64 kbps

- '2' : 80 kbps
- '3' : 96 kbps
- '4' : 128 kbps
- '5' : 256 kbps

d) Example

```
var value:Number = ext_fscommand2("GetRecLineInbps");
```

3.8.2 GetRecVoicebps

a) Explanation of Commands : Return the currently established MIC Volume Value

b) Input Value : None

c) Return Value : '0 ~ 5'

- '0' : 32 kbps
- '1' : 64 kbps
- '2' : 80 kbps
- '3' : 96 kbps
- '4' : 128 kbps
- '5' : 256 kbps

d) Example

```
var value:Number = ext_fscommand2("GetRecVoicebps");
```

3.8.3 GetRecFMbps

a) Explanation of Commands : The currently established FM Radio Quality Value

b) Input Value : None

c) Return Value : '0 ~ 5'

- '0' : 32 kbps
- '1' : 64 kbps
- '2' : 80 kbps
- '3' : 96 kbps
- '4' : 128 kbps
- '5' : 256 kbps

d) Example

```
var value:Number = ext_fscommand2("GetRecFMbps");
```

3.8.4 GetRecLineInVolume

a) Explanation of Commands : Return the currently established Line-in Volume Value

b) Input Value : None

c) Return Value : '0 ~ 4'

- '0' : Volume 1
- '1' : Volume 2
- '2' : Volume 3
- '3' : Volume 4
- '4' : Volume 5

d) Example

```
var value:Number = ext_fscommand2("GetRecLineInVolume");
```

3.8.5 GetRecVoiceVolume

a) Explanation of Commands: Return the currently established MIC Volume Value

b) Input Value : None

c) Return Value : '0 ~ 4'

- '0' : Volume 1

- '1' : Volume 2
- '2' : Volume 3
- '3' : Volume 4
- '4' : Volume 5

d) Example

```
var value:Number = ext_fscommand2("GetRecVoiceVolume");
```

3.8.6 GetRecAutoSync

a) Explanation of Commands : Return the currently established Auto Sync Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Off
- '1' : On

d) Example

```
var value:Number = ext_fscommand2("GetRecAutoSync");
```

3.8.7 GetRecMode

a) Explanation of Commands : Return the currently established Rec Mode Value b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Line-in Recording
- '1' : MIC Recording

d) Example

```
var value:Number = ext_fscommand2("GetRecMode");
```

3.8.8 GetRecCurTime

a) Explanation of Commands : Return Recording Time Value when the recording is currently activated or stopped temporarily. But other than that, return value is '0'.

b) Input Value : None

c) Return Value : '0 ~'

- '0 ~' : Sec (unit)

d) Example

```
var value:Number = ext_fscommand2("GetRecCurTime");
```

3.8.9 GetRecTotalTime

a) Explanation of Commands : Return the max possible record time when recording is being activated or is stopped temporarily. Other than that, the return value is '0'..

b) Input Value : None

c) Return Value : '0 ~'

- '0 ~' : Sec (unit)

d) Example

```
var value:Number = ext_fscommand2("GetRecTotalTime");
```

3.8.10 GetRecFileName

a) Explanation of Commands : If the current motion is recording or temporarily stopping recording, the recording file name is saved and in other cases, save the latest recorded file name is saved in the variables.

The latest recorded file name might be changed by the type of recording('GetRecMode').

b) Input Value : String Type Variable Name

c) Return Value : '-1', '1'

- '-1' : When it fails to work normally because there's no recent recorded files,
'No File' is saved in Variable

- '1' : When it normally saves the file name in Variable

d) Example

```
var recordName:String;
```

```
var value:Number = ext_fscommand2("GetRecFileName", "recordName");
```

e) Example Explanation : If Line-in recording is working, '1' is saved in Value, in 'recordName', ENR file name, for an example "ENR 2009-01-02 16-00-15.wma", is saved in 'recordName'

3.9 FM Radio Mode Get Parameter FS Command

Explanation of the FS Command that reads FM Radio Mode Value. FM Radio Free set is controlled by the name of 'Radio.ini' File in 'System' Folder.

3.9.1 GetFmrStereo

a) Explanation of Commands : Return the currently established FM Radio Stereo Value.

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Mono

- '1' : Stereo

d) Example

```
var value:Number = ext_fscommand2("GetFmrStereo");
```

3.9.2 GetFmrRegion

a) Explanation of Commands : Return the currently established FM Radio Local Value

b) Input Value : None

c) Return Value : '0 ~ 3'

- '0' : China and Europe

- '1' : Japan

- '2' : Korea

- '3' : US

d) Example

```
var value:Number = ext_fscommand2("GetFmrRegion");
```

3.9.3 GetFmrMinFreq

a) Explanation of Commands : Return the the Minimum Radio Frequency as the currently established FM Radio Local Setting b) Input Value : None

c) Return Value : '76000', '87500'

- '76000' : When it is set as Japan (unit) kHz

- '87500' : When it is set other than Japan (unit) kHz

d) Example

```
var value:Number = ext_fscommand2("GetFmrMinFreq");
```

3.9.4 GetFmrMaxFreq

a) Explanation of Commands : Return the maximum FM Radio Frequency Value

b) Input Value : None

c) Return Value : '118000'

- '118000' : (unit) kHz

d) Example

```
var value:Number = ext_fscommand2("GetFmrMaxFreq");
```

3.9.5 GetFmrCurrentFreq

a) Explanation of Commands : Return the currently established FM Radio Frequency Value

b) Input Value : None

c) Return Value : '76000 ~ 118000'

- '76000 ~ 118000' : (unit) kHz

d) Example

```
var value:Number = ext_fscommand2("GetFmrCurrentFreq");
```

3.9.6 GetFmrPresetOn

a) Explanation of Commands : Return the currently established FM Radio Preset Value

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Off

- '1' : On

d) Example

```
var value:Number = ext_fscommand2("GetFmrPresetOn");
```

3.9.7 GetFmrPresetIndex

a) Explanation of Commands : Return the currently established FM Radio Preset Index

b) Input Value : None

c) Return Value : '0 ~ 23'

d) Example

```
var value:Number = ext_fscommand2("GetFmrPresetIndex");
```

3.9.8 GetFmrNonePresetIndex

a) Explanation of Commands : Return the First Preset Index Value among the Preset Indexes that aren't given FM Radio Frequency.

b) Input Value : None

c) Return Value : '-1 ~ 23'

- '-1' : When every Preset is given a Channel

- '0 ~ 23' : When it works normally

d) Example

```
var value:Number = ext_fscommand2("GetFmrNonePresetIndex");
```

3.9.9 GetFmrPresetFreq

a) Explanation of Commands : Return the Inputted Index's FM Radio Frequency Value

b) Input Value : Preset Index

c) Return Value : '-1', '76000' ~ '118000'

- '-1' : When the inputted index doesn't have FM Radio Frequency, or when inputted index exceeds the limit.

- '76000' ~ '118000' : (unit) kHz

d) Example

```
var value:Number = ext_fscommand2("GetFmrPresetFreq", "5");
```

e) Example Explanation

When the fifth Index Frequency is saved in 91.9Mhz, the Value Variable has '91900'

3.9.10 GetFmrNearPresetIndex

- a) Explanation of Commands : Return the closest FM Preset Index to FM Radio Frequency
- b) Input Value : FM Radio Frequency. (unit) kHz
- c) Return Value : '0 ~ 23'
- d) Example

```
var value:Number = ext_fscommand2("GetFmrNearPresetIndex", "91900");
```

3.9.11 GetFmrPresetName

- a) Explanation of Commands : Save FM Radio Preset Name in Variable
- b) Input Value : Preset Index. The string type variable where the name is to be saved
- c) Return Value : '-1 ~ 1'
 - '-1' : When the inputted index exceeds the limit
 - '0' : When the inputted index doesn't have the radio station name
 - '1' : When the radio station name is normally saved in Variable

d) Example

```
var radioStation:String;  
var value:Number = ext_fscommand2("GetFmrPresetName", "5", "radioStation");
```

e) Example Explanation

When "Station 1" is saved at the fifth in FM Radio Preset list, "Station 1" is saved in 'RadioStation' Variable.

3.9.12 GetFmrScanState

- a) Explanation of Commands : Return the value of FM Radio Scan State
- b) Input Value : None
- c) Return Value : '-1' '1'
 - '-1' : No Scan
 - '1' : Scanning

d) Example

```
var value:Number = ext_fscommand2("GetFmrScanState");
```

3.10 Mobile TV(DMB/DAB) Mode Get Parameter FS Command

Mobile TV Mode Get Parameter FS Command is going to be released

3.11 Etc. Get Parameter FS Command

The explanation of FS Command that reads ETC UCI Value

3.11.1 GetEtcUIConfig

- a) Explanation of Commands : In each Flash, there are times when a save function is needed without necessarily controlling the Low Level class, as in the GUI settings. For times like these, the S9 provides a function to manage UI values in strings. If the 'PARAM.CFG' file in the 'System' folder as in formatting the device, or the settings values become initialized through the 'Restore System Default' function ('SetSysDefault'), the strings are initialized to "00\00". Also, because of the nature of a UCI, the variable information can often be changed due to the variety of UCI's from different developers. Hence, when saving UI values through this variable, there must be exception handling in case the value is incorrect, such as by attaching a unique header.

Please note that the base UI uses "number of settings|value1|value2..." and initializes if the 'number of settings' is different from what is in the ActionScript.

b) Input Value : '0 ~ 15', String Type Variable Name

First input value

- '0' : Music
- '1' : Video
- '2' : Radio
- '3' : Record
- '4' : MobileTV
- '5' : Flash
- '6' : Document
- '7' : Picture
- '8' : Dictionary
- '9' : Util
- '10' : Etc.
- '11' : Mainmenu
- '12' : Mainmenu2
- '13' : Mainmenu3
- '14' : Setting
- '15' : Browser

Second Input Value

- String Type Variable Name from Low Level Class

c) Return Value : '-1' '1'

- '-1' : When it fails to work normally since Input Value exceeds the limit
- '1' : When it works normally

d) Example

```
var musicData:String;  
var value:Number = ext_fscommand2("GetEtcUIConfig", "0", "musicData");
```

e) Example Explanation

If three pre-existing settings values are used in the music UI at 10, 20 and 100 respectively, it returns with the 'value' variable at '1' and the 'musicData' variable with a '3|10|20|100' string.

3.11.2 GetEtcSettingString

a) Explanation of Commands: An FS Command used in the settings menu, it takes the list name and values that belong to the current step, converts them with the "|" separator, saves them in the variable, and then

returns the total number of lists.

b) Input Value : String type variable that is to save the list

c) Return Value : '0 ~'

- '0 ~' : Return the total number of lists in the current set menu

d) Example

```
var listName:String;  
var listValue:String;  
var value:Number = ext_fscommand2("GetEtcSettingString", "listName", "listValue");
```

e) Example Explanation

If the current step is the recording menu, '6' is saved in the 'value', and "Line-In Quality|Mic Quality|FM Quality|Line-In Volume|Mic Volume|Auto Sync" in the listName variable. In the listValue variable, "128 k|64 K|128 k|5|5|Off" is saved.

3.11.3 GetEtcSettingCurrIndex

- a) Explanation of Commands : Return the selected Index Value
- b) Input Value : None
- c) Return Value : '0 ~'
 - '0 ~' : Return the selected Index Value

d) Example

```
var value:Number = ext_fscommand2("GetEtcSettingCurrIndex");
```

3.11.4 GetEtcSettingStep

- a) Explanation of Commands : Return the current step value. The main menu has total three steps
- b) Input Value : None
- c) Return Value : '0 ~ 2'
 - '0' : Step for 'JetEffect 2.0', 'Display'
 - '1' : In Step '0', displaying the value of "Language", "Fonts", "Playtime"
 - '2' : In Step '1', displaying 'System Fonts', 'User Fonts' when choosing 'Fonts'

d) Example

```
var value:Number = ext_fscommand2("GetEtcSettingStep");
```

3.11.5 GetEtcSettingTitle

- a) Explanation of Commands : Save the string to be displayed at the top of the current step in Variable
- b) Input Value : String Type Variable Name
- c) Return Value : '1'
 - '1' : Return '1'

d) Example

```
var settingTitle:String;  
var value:Number = ext_fscommand2("GetEtcSettingTitle", "settingTitle");
```

- e) Example Explanation : Select 'Display' in step '0'. If the current step is 1, the 'Display' string is saved in the 'settingTitle' variable and '1' in the 'value' variable.

3.11.6 GetEtcVolume

- a) Explanation of Commands : Return the current Volume Value
- b) Input Value : None
- c) Return Value : '0 ~ 40'
 - '0 ~ 40' : Return the current Device's Volume Value

d) Example

```
var value:Number = ext_fscommand2("GetEtcVolume");
```

3.11.7 GetEtcTotalPLNum

- a) Explanation of Commands : An FS Command used in the Music, Video, and Picture Mode, this command returns the total number of the playlists (Music, Video) or of the picture lists (Picture). Whenever a file is selected in the browser or the play boundary is changed due to 'SetAudBoundary' FS Command, the 'GetEtcTotalPLNum' value must be read and reflected in the GUI.

b) Input Value : None

c) Return Value : '-1 ~'

- '-1' : When the current Mode is not Music, Video, Picture

- '0 ~' : Return the number of the current lists

d) Example

```
var value:Number = ext_fscommand2("GetEtcTotalPLNum");
```

3.11.8 GetEtcCurPLIndex

a) Explanation of Commands : An FS Command used in the Music, Video, and Picture Mode, this command returns the index value that is currently being played back or set on the playlists (Music, Video) or the picture lists (Picture). The index value ranges from '0' to 'Total number of playlists -1'. Whenever a file is selected in the browser or the song in play changes through Next Song etc., the 'GetEtcCurPLIndex' value must be read and reflected in the GUI.

b) Input Value : None

c) Return Value : '-1 ~'

- '-1' : When the current Mode is not Music, Video, Picture or there's no file in Playlist

- '0 ~' : Return the currently established list's Index Value

d) Example

```
var value:Number = ext_fscommand2("GetEtcCurPLIndex");
```

3.11.9 GetEtcFileName

a) Explanation of Commands : Saves the name of the file belonging to the index that is entered and returns the entered index value. This operates according to the playlist in the Music or the Video Mode, the picture list in the Picture Mode, and the most recently selected file in the browser in the Text or the Flash Mode.

b) Input Value : Index ('0' ~ 'The total number of lists - 1'), String Type Variable Name

c) Return Value : '-1 ~'

- '-1' : When there's no files in the playlist or the inputted index exceeds the limit

- '0 ~' : Return the inputted index value

d) Example

```
var fileName:String;
```

```
var value:Number = ext_fscommand2("GetEtcFileName", "5", "fileName");
```

e) Example Explanation : If there are 10 songs in total on the playlist and the name of the file that belongs to index 5 is 'iAUDIO – Friends & Lovers.mp3', the 'iAUDIO – Friends & Lovers.mp3' string is saved in the 'fileName' variable and '5' in the 'value' variable.

3.11.10 GetEtcState

a) Explanation of Commands : Returns the current state of the device. In the Low Level class, this command controls according to the predefined state in each of the modes. Since this command supports multi-tasking, the current state of the device must be read in the ActionScript through this FS Command to display the corresponding GUI. For example, if the user views a document during music playback and then goes back to music, the state value is '1', and the GUI that corresponds to '1' must be displayed.

b) Input Value : None

c) Return Value : '0 ~ 11'

- '0' : Undefined(When Mode hasn't been change since booting)

- '1' : Play(Music, Video Mode)

- '2' : Pause(Music, Video Mode)

- '3' : Stop(Music, Video Mode)

- '4' : Radio(Radio Mode)

- '5' : Record(Radio Mode)
- '6' : Reserve
- '7' : Record(Record Mode)
- '8' : Record_Pause(Record Mode)
- '9' : Stop(Record Mode)
- '10' : Play(Record Mode)
- '11' : Play_Pause(Record Mode)
- '12' : Mobile TV(Mobile TV Mode)
- '13' : Record(Mobile TV Mode)

d) Example

```
var value:Number = ext_fscommand2("GetEtcState");
```

3.11.11 GetEtcFavorite

a) Explanation of Commands : Returns the value of whether the current playlist is a favorite. If it is a favorite playlist, it must be processed through 'ext_fscommand2 ("EtcBrwSetInitialization", "Favorite")' when entering the browser. This is because, when a file is selected in the browser, the playlist is derived from the folder which contains the file in question but, when a file is selected on the favorites list, the favorites list is processed as a playlist. Only the Music or Video Mode which has a favorites list is distinguished.

Other modes always return '0'.

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : Normal Playlist
- '1' : Favorite Playlist

d) Example

```
var value:Number = ext_fscommand2("GetEtcFavorite");
```

3.11.12 GetEtcTASValue

a) Explanation of Commands: Returns the acceleration sensor value. Whenever the value of the acceleration sensor is changed, 'F11' key is triggered. When it is triggered, the ActionScript uses this FS Command to obtain the acceleration sensor value, which allows for a change to an appropriate GUI.

Furthermore, because there are times when the Flash Engine class does not operate, the developer must obtain the acceleration sensor value with this FS Command and change the GUI in the below cases.

- When 'F11' Key is triggered : The value 'GetEtcTASValue' is read and the GUI is processed accordingly.
- When 'F12' Key is triggered : 'F12' Key is triggered when AMOLED switches from Off to On, or when beginning playback. In this case, the Hold status should be read first (through the "GetSysHoldKey" FS Command) and, if the Hold is Off, the value from 'GetEtcTASValue' should be read and the GUI processed accordingly.
- When 'Scroll Lock' Key is triggered : The 'GetSysHoldKey' FS Command should be read and, if the Hold has been switched from On to Off, the value from 'GetEtcTASValue' should be read and the GUI processed accordingly.

b) Input Value : None

c) Return Value : '0 ~ 3'

- '0' : Regular Direction

- '1' : Turn clockwise 90 degrees
- '2' : Turn clockwise 90 degrees
- '3' : Turn clockwise 90 degrees

d) Example

```
var value:Number = ext_fscommand2("GetEtcTASValue");
```

3.11.13 GetEtcOpenState

a) Explanation of Commands : Returns the playback attempt outcome value when the playback FS Command ('KeyAudPlay') is triggered, or the current song has finished playing and the Low Level class moves to the next song. There are times when the playback attempt fails for reasons such as files that are not supported in the normal user environment or expired DRM. When this happens, the 'F12' key is triggered in the Low Level class. When an 'F12' key event occurs, the ActionScript can read this FS Command to confirm the playback attempt failure in order to display the GUI accordingly. This FS Command, unlike other FS Commands, has the Read & clear attribute, which causes it to internally initialize to '0' (normal state) upon being called. The variable saves a string regarding the current state.

b) Input Value : String Type Variable Name

c) Return Value : '0 ~ 10'

- '0' : 'Normal Playback'
- '1' : 'Wrong File'
- '2' : 'Unsupportive file'
- '3' : 'DCF Error'
- '4' : 'DCF Expiration'
- '5' : 'Unregistered MP3P'
- '6' : 'Unauthorized'
- '7' : 'Time Information Error'
- '8' : 'Expired License'
- '9' : 'Expired Period'
- '10' : 'The termination of Usage'

d) Example

```
var openState:String;
var value:Number = ext_fscommand2("GetEtcOpenState", "openState");
```

e) Example Explanation : When it normally plays, 'Play Normal' is saved in 'openState' Variable and '0' is saved in 'Value' Variable

3.11.14 GetEtcTASValueX

a) Explanation of Commands: Returned the accelerating sensor value on 'X' axis.

The sensor is located in the middle of the player and the 'X' axis is separated to (+)and (-)direction. (+) direction : same toward the location of playing button., (-) direction is headed to the charging input and power button. When the player is paused, the value from X axis would be returned. The values could have small different errors from each player and it depends on the surrounded system environment.

b) Input Value : None

c) Return Value : '-512 ~ 511'

d)) Example

```
var value:Number = ext_fscommand2("GetEtcTASValueX");
```

3.11.15 GetEtcTASValueY

a) Explanation of Commands: Return the accelerating sensor value from Y axis. The sensor is located in the middle of the player and the Y axis is separated to (+) direction; from front side to right, (-) direction; from front side to left. When the player is paused, the value from Y axis would be returned. The values could have small different errors from each player and it depends on the surrounded system environment.

b) Input Value : None

c) Return Value : '-512 ~ 511'

d) Example

```
var value:Number = ext_fscommand2("GetEtcTASValueY");
```

3.11.16 GetEtcTASValueZ

a) Explanation of Commands: Return the accelerating sensor value from Z axis. The sensor is located in the middle of the player and the axis is separated to (+) direction; the front side (LCD location), (-) direction; the location of MIC, reset button. When the player is paused, the value from Z axis would be returned. The values could have small different errors from each player and it depends on the surrounded system environment.

b) Input Value : None

c) Return Value : '-512 ~ 511'

d) Example

```
var value:Number = ext_fscommand2("GetEtcTASValueZ");
```

4 Set Parameter FS Command

4.1 JetEffect 2.0 Set Parameter FS Command

The explanation of FS Command that sets JetEffect 2.0

4.1.1 SetJetEffectIndex

a) Explanation of Commands : Set JetEffect 2.0's Index Value. According to the set, the Playsound of Music or Video changes.

b) Input Value : '0 ~ 29'

- '0' : 'User 1'

- '1' : 'User 2'

- '2' : 'User 3'

- '3' : 'User 4'

- '4' : 'Normal'

- '5' : 'BBE'

- '6' : 'BBE VIVA'

- '7' : 'BBE VIVA 2'

- '8' : 'BBE Mach3Bass'

- '9' : 'BBE MP'

- '10' : 'BBE Headphone'

- '11' : 'BBE Headphone 2'

- '12' : 'BBE Headphone 3'

- '13' : 'Rock'

- '14' : 'Jazz'

- '15' : 'Classic'
- '16' : 'Ballad'
- '17' : 'Pop'
- '18' : 'Club'
- '19' : 'Funk'
- '20' : 'Hip Hop'
- '21' : 'Techno'
- '22' : 'Blues'
- '23' : 'Metal'
- '24' : 'Dance'
- '25' : 'Rap'
- '26' : 'Wide'
- '27' : 'X-Bass'
- '28' : 'Hall'
- '29' : 'Vocal'

c) Return Value : '-1 ~ 29'

- '-1' : When it fails to work due to that 'Input Value' exceeds the limit
- '0 ~ 29' : Return 'Input Value' when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetJetEffectIndex", "4");
```

e) Example Explanation : Return '4' in 'value' , Set JetEffect 2.0 with 'Normal'

4.1.2 SetJetUserIndex

a) Explanation of Commands : Set the applied User Preset Index when sets JetEffect 2.0 User Preset

b) Input Value : '0 ~ 3'

- '0' : 'User 1'
- '1' : 'User 2'
- '2' : 'User 3'
- '3' : 'User 4'

c) Return Value : '-1 ~ 3'

- '-1' : When the Input Value exceeds the limit
- '0 ~ 3' : Return 'Input Value' when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetJetUserIndex", "0");
```

e) Example Explanation : Returns '0' for the 'value' variable, sets the JetEffect 2.0 User Preset to 'User 1'.

Afterwards, detailed settings such as 'SetJetUserBBE' and 'SetJetUserM3B' change 'User 1' preset values.

4.1.3 SetJetUserBBE

a) Explanation of Commands : Change the BBE of JetEffect User Index set by 'SetJetUserIndex'
FS Command

b) Input Value : '0 ~ 10'

- '0' : BBE Off
- '1 ~ 10' : BBE Value

c) Return Value : '-1 ~ 10'

- '-1' : Input Value exceeds the limit
- '0 ~ 10' : Return the Input Value

d) Example

```
var value:Number = ext_fscommand2("SetJetUserBBE", "1");
```

e) Example Explanation : Return '1' to 'Value' Variable, Set '1' as the JetEffect 2.0 User Preset BBE Value

4.1.4 SetJetUserM3B

- a) Explanation of Commands : Change the Mach3Bass of JetEffect User Index set by 'SetJetUserIndex'
FS Command
- b) Input Value : '0 ~ 10'
- '0' : Mach3Bass Off
- '1 ~ 10' : Mach3Bass Value
- c) Return Value : '-1 ~ 10'
- '-1' : When Input Value exceeds the limit
- '0 ~ 10' : Return the Input Value
- d) Example
`var value:Number = ext_fscommand2("SetJetUserM3B", "2");`
- e) Example Explanation : Return '2' to 'Value' Variable, Set '2' as the JetEffect 2.0 User Preset Mach3Bass Value

4.1.5 SetJetUser3DS

- a) Explanation of Commands : Change the 3D Surround of JetEffect User Index set by 'SetJetUserIndex'
FS Command
- b) Input Value : '0 ~ 10'
- '0' : 3D Surround Off
- '1 ~ 10' : 3D Surround Value
- c) Return Value : '-1 ~ 10'
- '-1' : When 'Input Value' exceeds the limit
- '0 ~ 10' : Return the Input Value
- d) Example
`var value:Number = ext_fscommand2("SetJetUser3DS", "3");`
- e) Example Explanation : Return '3' to "Value" Variable, Set '3' as the JetEffect 2.0 User Preset Mach3Bass Value

4.1.6 SetJetUserMPE

- a) Explanation of Commands : Change the MP Enhance of JetEffect User Index set by 'SetJetUserIndex'
FS Command
- b) Input Value : '0 ~ 1'
- '0' : MP Enhance Off
- '1' : MP Enhance On
- c) Return Value : '-1 ~ 1'
- '-1' : When 'Input Value' exceeds the limit
- '0 ~ 1' : Return the Input Value
- d) Example
`var value:Number = ext_fscommand2("SetJetUserMPE", "0");`
- e) Example Explanation : Return '0' to 'Value' Variable, Set 'Off' as the JetEffect 2.0 User Preset MP Enhance

4.1.7 SetJetUserSTE

- a) Explanation of Commands : Change the Stereo Enhance of JetEffect User Index set by 'SetJetUserIndex'
FS Command
- b) Input Value : '0 ~ 10'
- '0' : Stereo Enhance Off

- '1 ~ 10' : Stereo Enhance Value
- c) Return Value : '-1 ~ 10'
 - '-1' : When 'Input Value' exceeds the limit
 - '0 ~ 10' : Return the Input Value
- d) Example


```
var value:Number = ext_fscommand2("SetJetUserSTE", "4");
```
- e) Example Explanation : Return '4' to "Value" Variable, Set '4' as the JetEffect 2.0 User Preset Stereo Enhance Value

4.1.8 SetJetUserEQBandGain

- a) Explanation of Commands : Set the 'N'th Band EQ of JetEffect User Index established by 'SetJetUserIndex' FS Command
- b) Input Value : '0 ~ 4', '0 ~ 24'
 - First input value
 - '0' : Band 1
 - '1' : Band 2
 - '2' : Band 3
 - '3' : Band 4
 - '4' : Band 5
 - Second input value
 - '0' : -12 dB
 - '12' : 0 dB
 - '24' : 12 dB
- c) Return Value : '-1 ~ 24'
 - '-1' : When 'Input Value' exceeds the limit
 - '0 ~ 24' : Return the Gain Value
- d) Example


```
var value:Number = ext_fscommand2("SetJetUserEQBandGain", "0", "16");
```
- e) Example Explanation : Return '16' to "Value" Variable, Set '4 dB' as the JetEffect 2.0 User Preset 1st Band EQ Gain

4.1.9 SetJetUserEQCenterFreq

- a) Explanation of Commands : Set the 'N'th Band EQ Center Frequency of JetEffect User Index established by 'SetJetUserIndex' FS Command
- b) Input Value : '0 ~ 4', '0 ~ 3'
 - First Input Value
 - '0' : Band 1
 - '1' : Band 2
 - '2' : Band 3
 - '3' : Band 4
 - '4' : Band 5
 - Second Input Value
 - Band 1
 - '0' : 80 Hz
 - '1' : 105 Hz
 - '2' : 135 Hz
 - '3' : 175 Hz
 - Band 2
 - '0' : 220 Hz

'1' : 300 Hz

'2' : 385 Hz

'3' : 500 Hz

- Band 3

'0' : 650 Hz

'1' : 780 Hz

'2' : 1.1 kHz

'3' : 1.4 kHz

- Band 4

'0' : 1.8 kHz

'1' : 2.4 kHz

'2' : 3.0 kHz

'3' : 4.1 kHz

- Band 5

'0' : 6.9 kHz

'1' : 9 kHz

'2' : 11.7 kHz

'3' : 13 Hz

c) Return Value : '-1 ~ 3'

- '-1' : When 'Input Value' exceeds the limit

- '0 ~ 3' : Return the inputted Center Frequency Value

d) Example

```
var value:Number = ext_fscommand2("SetJetUserEQCenterFreq", "1", "3");
```

e) Example Explanation : Return '3' to "Value" Variable, Set '500 Hz' as the JetEffect 2.0 User Preset 2 Band EQ Center Frequency Value

4.1.10 GetJetUserEQBandWidth

a) Explanation of Commands : Set the 'N'th Band EQ Bandwidth of JetEffect User Index established by 'SetJetUserIndex' FS Command

b) Input Value : '0 ~ 4', '0 ~ 2'

First Input Value

- '0' : Band 1

- '1' : Band 2

- '2' : Band 3

- '3' : Band 4

- '4' : Band 5

Second Input Value

- '0' : Narrow

- '1' : Normal

- '2' : Wide

c) Return Value : '-1 ~ 2'

- '-1' : When 'Input Value' exceeds the limit

- '0 ~ 2' : Return the imputed Bandwidth Value

d) Example

```
var value:Number = ext_fscommand2("GetJetUserEQBandWidth", "2", "0");
```

e) Example Explanation : Return '0' to "Value" Variable, Set 'Narrow' as the JetEffect 2.0 User Preset 3 Band EQ Center Bandwidth Value

4.1.11 SetJetSettingMode

- a) Explanation of Commands : An FS Command used in the JetEffect 2.0 user menu, this command controls the settings menu. This FS Command must be configured properly to process multiple languages in the JetEffect 2.0 user settings menu.
- b) Input Value : '0 ~ 2'
 - '0' : EQ
 - '1' : BBE+
 - '2' : STE
- c) Return Value : '-1 ~'
 - '-1' : When 'Input Value' exceeds the limit
 - '0 ~' : Return the number of Setting Menu
- d) Example

```
var value:Number = ext_fscommand2("SetJetSettingMode", "1");
```
- e) Example Explanation : Return '4' to 'value' Variable. And 'BBE', 'Mach3Bass', '3D Surround', 'MP Enhance' are able to be set in BBE+ Setting Menu.

4.2 Display Set Parameter FS Command

Explaining Display Set and FS Command

4.2.1 SetDisLanguage

- a) Explanation of Commands: Set the language
- b) Input Value : '0 ~ 14'
 - '0' : English
 - '1' : Korean
 - '2' : Czech
 - '3' : Dutch
 - '4' : French
 - '5' : German
 - '6' : Hungarian
 - '7' : Italian
 - '8' : Japanese
 - '9' : Polish
 - '10' : Russian
 - '11' : Chinese(Modern)
 - '12' : Chinese(Traditional)
 - '13' : Spanish
 - '14' : Swedish
- c) Return Value : '-1 ~ 14'
 - '-1' : When 'Input Value' exceeds the limit
 - '0 ~ 14' : Return 'Input Value'
- d) Example

```
var value:Number = ext_fscommand2("SetDisLanguage", "1");
```
- e) Example Explanation : Return '1' to 'value' Variable, Function with 'Korean'

4.2.2 SetDisFont

- a) Explanation of Commands : Set Font
- b) Input Value : System Font
 - '1' : User Font

c) Return Value : '-1 ~ 1'

- '-1' : When 'Input Value' exceeds the limit
- '0 ~ 1' : Return the input value

d) Example

```
var value:Number = ext_fscommand2("SetDisFont", "0");
```

e) Example Explanation : Return '0' to 'value' Variable, Function with 'System Font'

4.2.3 SetDisplayTime

a) Explanation of Commands : Setting the played time. For your information, if the set is the remaining time, the return time will be 'the total time' minus 'the current played time'.

b) Input Value : '0 ~ 1'

- '0' : Played Time
- '1' : Remaining Time

c) Return Value : '-1 ~ 1'

- '-1' : When 'Input Value' exceeds the limit
- '0 ~ 1' : Return the Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetDisplayTime", "0");
```

e) Example Explanation : Return '0' to 'value' Variable, Function with 'Played Time'

4.2.4 SetDisLyrics

a) Explanation of Commands : Determines whether to display lyrics during music playback. If the lyrics display is set to 'Off', the LDB type of the song that is currently playing which is obtained through the 'GetAudLDBType' FS Command is always 0, signifying 'No LDB'.

b) Input Value : '0 ~ 1'

- '0' : Off
- '1' : On

c) Return Value : '-1 ~ 1'

- '-1' : When 'Input Value' exceeds the limit
- '0 ~ 1' : Return the Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetDisLyrics", "1");
```

e) Example Explanation : Return '1' to 'value' Variable, Function with Lyric 'On'

4.2.5 SetDisLCDAutoOff

a) Explanation of Commands : Set Automatic Display Off

b) Input Value : '0 ~ 7'

- '0' : Off
- '1' : 5 sec
- '2' : 10 sec
- '3' : 20 sec
- '4' : 30 sec
- '5' : 40 sec
- '6' : 50 sec
- '7' : 60 sec

c) Return Value : '-1 ~ 7'

- '-1' : When 'Input Value' exceeds the limit
- '0 ~ 7' : Return the Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetDisLCDAutoOff", "2");
```

e) Example Explanation : Return '2' to 'value' Variable and the display automatically turns off 10 sec after pressing key or touch.

4.2.6 SetDisBrightness

a) Explanation of Commands : Set Brightness

b) Input Value : '0 ~ 5'

c) Return Value : '-1 ~ 5'

- '-1' : When 'Input Value' exceeds the limit

- '0 ~ 5' : : Return the Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetDisBrightness", "5");
```

e) Example Explanation : Return '5' to 'value' Variable and set the brightness '5'

4.2.7 GetDisTVOutMode

a) Explanation of Commands : Set Display TV Out Mode

b) Input Value : '0 ~ 1'

- '0' : NTSC

- '1' : PAL

c) Return Value : '-1 ~ 1'

- '-1' : When Input Value exceeds the limit

- '0 ~ 1' : Return Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("GetDisTVOutMode", "0");
```

e) Example Explanation : Return '0' to 'value' Variable and set TV out as 'NTSC'

4.2.8 SetDisWallpaper

a) Explanation of Commands : Select to enable/disable wallpaper. When the FS Command that selects the wallpaper is called, input value will be set as '1' in the firmware.

b) Input Value : '0 ~ 1'

- '0' : Normal background

- '1' : Wallpaper

c) Return Value : '-1 ~ 1'

- '-1' : When not processed normally(ex. Input Value out of range)

- '0 ~ 1' : Input value returned when processed normally.

d) Example

```
var value:Number = ext_fscommand2("SetDisWallpaper", "1");
```

e) Example explanation : '1' is returned for 'value' variable, and wallpaper will be used if already designated.

4.2.9 SetDisLCDAutoOffReset

a) Explanation of Commands : Reset the LCD timer to the value set through 'SetDisLCDAutoOff'. LCD timer automatically resets when there is key input or touch pad input in Low Level. When the same function is needed in ActionScript, this FS Command

can be used to reset the LCD timer.

b) Input Value : None

c) Return Value : '-1, 1'

- '-1' : When not processed normally
- '1' : value returned when processed normally.

d) Example

```
var value:Number = ext_fscommand2("SetDisLCDAutoOffReset");
```

4.3 Timer Get Parameter FS Command

The explanation of FS Commands for 'Time' and 'Alarm'

4.3.1 SetTimTime

a) Explanation of Commands: Set Time.

b) Input Value :

- First Input Value : Year (Range 2000 ~ 2030)
- Second Input Value : Month(Range 0 ~ 11)
- Third Input Value : Day(Range 1 ~ 31)
- Fourth Input Value : Time(Range 0 ~ 23)
- Fifth Input Value : Minute(Range 0 ~ 59)
- Sixth Input Value : Second(Range 0 ~ 59)

c) Return Value : '-1', '1'

- '-1' : When Input Value exceeds the limit
- '1' : When it works normally

d) Example

```
var value:Number = ext_fscommand2("SetTimTime", "2009", "1", "2", "20", "18", "35");
```

e) Example Explanation : Return '1' to 'value' Variable, and set, '2009 Year 2 Month 2 Date 20 Hours 18 Minutes 35 Second

4.3.2 SetTimAlarmDuration

a) Explanation of Commands: Set Alarm Duration

b) Input Value : '0 ~ 6'

- '0' : Always On
- '1' : 20 min
- '2' : 40 min
- '3' : 60 min
- '4' : 80 min
- '5' : 100 min
- '6' : 120 min

c) Return Value : '-1 ~ 6'

- '-1' : When input Value exceeds the limit
- '0 ~ 6' : Return Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetTimAlarmDuration", "1");
```

e) Example Explanation : Return '1' to 'value' Variable, and set the alarm duration by '20 min'

4.3.3 SetTimAlarmRepeat

a) Explanation of Commands : Set Alarm Repeat

b) Input Value : '0 ~ 1'

- '0' : Once
- '1' : Repeat

c) Return Value : '-1 ~ 1'

- '-1' : When Input Value exceeds the limit
- '0 ~ 1' : Return Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetTimAlarmRepeat", "0");
```

e) Example Explanation : Return '0' to 'value' Variable, and set the alarm repeat at 'Once'

4.3.4 SetTimAlarmMode

a) Explanation of Commands : Set Alarm Mode

b) Input Value : '0 ~ 5'

- '0' : Off
- '1' : Music
- '2' : FM Radio
- '3' : FM Radio Recording
- '4' : DMB/DAB(DMB only supported)
- '5' : DMB/DAB Recording(DMB only supported)

c) Return Value : '-1 ~ 5'

- '-1' : When Input Value exceeds the limit
- '0 ~ 5' : Return Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetTimAlarmMode", "2");
```

e) Example Explanation : 'Return '2' to value' Variable, and set the alarm as 'FM Radio'

4.3.5 SetTimAlarmTime

a) Explanation of Commands : Set Alarm Time

b) Input Value : '0 ~ 23', '0 ~ 59'

First Input Value

- '0 ~ 23' : hour

Second Input Value

- '0 ~ 59' : min

c) Return Value : '-1', '1'

- '-1' : When Input Value exceeds the limit
- '1' : When it works normally

d) Example

```
var value:Number = ext_fscommand2("GetTimAlarmTime", "10", "7");
```

e) Example Explanation : Returns '1' to the 'value' variable and sets the alarm to go off at '10 hours' and '7 minutes'. If the alarm time is before the current time, it is set to the next day. If the alarm time is after the current time, the alarm is set to that day. Please note that, if the alarm goes off when the device is on, the Low Level class adjusts the alarm settings to go off the next day and triggers an 'F12' event. The alarm occurrence can be confirmed by using the 'GetTimAlarmState' FS Command in the ActionScript.

4.3.6 SetTim24HDisplay

a) Explanation of Commands : Set Time Display. With FS Command, 'F12' event is activated to update GUI after saving the value in the Low level Class

- b) Input Value : '0 ~ 1'
 - '0' : Display 'AM / PM'
 - '1' : Display '24 hours'
- c) Return Value : '-1 ~ 1'
 - '-1' : When Input Value exceeds the limit
 - '0 ~ 1' : Return Input Value when it works normally
- d) Example


```
var value:Number = ext_fscommand2("SetTim24HDisplay", "0");
```
- e) Example Explanation : Return '0' to 'value' Variable

4.3.7 SetTimSleep

- a) Explanation of Commands : Set Time Sleep. The input value initializes to '0' when the device turns on.
- b) Input Value : '0 ~ 12'
 - '0' : Off
 - '1' : 10 min
 - '2' : 20 min
 - '3' : 30 min
 - '4' : 40 min
 - '5' : 50 min
 - '6' : 60 min
 - '7' : 70 min
 - '8' : 80 min
 - '9' : 90 min
 - '10' : 100 min
 - '11' : 110 min
 - '12' : 120 min
- c) Return Value : '-1 ~ 12'
 - '-1' : When Input Value exceeds the limit
 - '0 ~ 12' : Return the input value when it works normally
- d) Example


```
var value:Number = ext_fscommand2("SetTimSleep", "2");
```
- e) Example Explanation : Return '2' to 'value' Variable, and turn off the device after '20 min' with FS Command standard

4.3.8 SetTimAutoOff

- a) Explanation of Commands : Sets the automatic stand-by time. In the following cases, the device is considered to be in operation. If there is not device input in other circumstances, the device turns off after the certain amount of time that is set in the values.
 - Play Music
 - Play Video
 - Listen Radio
 - Watch DMB or DAB
 - Play Recording or Recorded File
 - When set Auto off timer at 'Off' with 'SetEtcLCDTimerOff' FS Command in Documents or Photo
 - When users press touch panel or input buttons, the timer initializes to the last input value.
- b) Input Value : '0 ~ 8'
 - '0' : Off

- '1' : 1 min
- '2' : 2 min
- '3' : 3 min
- '4' : 4 min
- '5' : 5 min
- '6' : 10 min
- '7' : 20 min
- '8' : 30 min

c) Return Value : '-1 ~ 8'

- '-1' : When Input Value exceeds the limit
- '0 ~ 8' : Return Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetTimAutoOff", "3");
```

e) Example Explanation : Return '3' to 'value' Variable, and turn off automatically when the device hasn't performed any functions for '3 min'

4.4 Bluetooth Set Parameter FS Command

The explanation of Bluetooth FS Command

4.4.1 SetBTHState

a) Explanation of Commands : Sets the Bluetooth power. Please note that, because Bluetooth takes a great operational volume and because of the electric current used in the Bluetooth chips, the battery life is shorter in comparison to operating normally.

b) Input Value : '0 ~ 1'

- '0' : Off
- '1' : On

c) Return Value : '-1 ~ 1'

- '-1' : When Input Value exceeds the limit
- '0 ~ 1' : Return the input value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetBTHState", "1");
```

e) Example Explanation : Return '1' to 'value' variable, and activating Bluetooth

4.4.2 SetBTHStartScan

a) Explanation of Commands : Start scanning Bluetooth Devices

b) Input Value : None

c) Return Value : '-1', '1'

- '-1' : When it fails to scan due to that the Bluetooth devices' power is turned off
- '1' : When it works normally

d) Example

```
var value:Number = ext_fscommand2("SetBTHStartScan");
```

4.4.3 SetBTHStopScan

a) Explanation of Commands : Stop scanning Bluetooth devices

b) Input Value : None

c) Return Value : '-1', '1'

- '-1' : When it fails to scan due to that the Bluetooth devices' power is turned off

- '1' : When it normally stops scanning

d) Example

```
var value:Number = ext_fscommand2("SetBTHStopScan");
```

4.4.4 SetBTHConnect

a) Explanation of Commands : Connects to the Bluetooth device that belongs to the index input.

If the Bluetooth device that belongs to the index input is already connected, the connection is terminated.

b) Input Value : '0 ~ 6'

- '0 ~ 6' : Index Value of a Bluetooth list

c) Return Value : '-1 ~ 1'

- '-1' : When the index exceeds the limit or the Bluetooth is turned off

- '0' : When disconnecting the existed Bluetooth by inputting new Bluetooth

- '1' : When it connects Bluetooth normally

d) Example

```
var value:Number = ext_fscommand2("SetBTHConnect", "0");
```

e) Example Explanation : If Bluetooth is on and a Bluetooth device is registered for index item 0 on the Bluetooth list, '1' is saved in the 'value' variable and the unit connects to the Bluetooth device assigned to index 0.

4.4.5 SetBTHRemoveList

a) Explanation of Commands : Remove the Bluetooth device from the Bluetooth list

b) Input Value : '0 ~ 6'

- '0 ~ 6' : Index Value of a Bluetooth list

c) Return Value : '-1', '1'

- '-1' : When it fails to remove the Bluetooth from the list due to that the Bluetooth is turned on

- '1' : When the Bluetooth is normally removed from the list

d) Example

```
var value:Number = ext_fscommand2("SetBTHRemoveList", "0");
```

e) Example Explanation : If Bluetooth is on and a Bluetooth device is registered for index item 0 on the Bluetooth list, '1' is saved in the 'value' variable and the index 0 item is removed from the Bluetooth list.

4.4.6 SetBTHPinCode

a) Explanation of Commands : Sets the bluetooth Pin Code in string input. Bluetooth Pin Code is composed of numbers. Therefore, the string being input must also be numbers. Default Bluetooth Pin Code is '0000'. The string value will be reset to '0000' when 1. 'PARAM.CFG' file in 'System' folder is deleted., 2. when settings are reset with 'Load Default' function('SetSysDefault').

b) Input Value : string with numbers from minimum 1 to maximum 15 digits.

c) Return Value : '-1', '1'

- '-1' : When not processed normally because there is no numbers or character other than numbers exist in the string.

- '1' : When set normally.

d) Example

```
var pinCode:String = "1234";
```

```
var value:Number = ext_fscommand2("SetBTHPinCode ", pinCode);
```

e) Example explanation : '1' value will be saved under value variable, and Bluetooth Pin Code will be changed to '1234'.

4.5 System Set Parameter FS Command

Explaining System setting FS Command

4.5.1 SetSysCtrlHoldState

- a) Explanation of Commands : Set Hold State which is used with Hold Key
- b) Input Value : '0 ~ 1'
 - '0' : All
 - '1' : Touch Screen only
- c) Return Value : '-1 ~ 1'
 - '-1' : When Input Value exceeds the limit
 - '0 ~ 1' : Return the input value when it works normally
- d) Example

```
var value:Number = ext_fscommand2("SetSysCtrlHoldState", "0");
```
- e) Example Explanation : Return '0' to 'value' Variable, and setting 'Hold All' with hold key

4.5.2 SetSysPan

- a) Explanation of Commands : Set Audio L/R output
- b) Input Value : '0 ~ 20'
 - '0 ~ 9' : Setting the right earphone output small. For instance, setting '0' minimizes the right earphone out
 - '10' : Right and Left have the same output
 - '11 ~ 20' : Setting the left earphone output small. For instance, setting '20' maximizes the right earphone out
- c) Return Value : '-1 ~ 20'
 - '-1' : When Input Value exceeds the limit
 - '0 ~ 20' : Return the input value when it works normally
- d) Example

```
var value:Number = ext_fscommand2("SetSysPan", "10");
```
- e) Example Explanation : Return '10' to 'value' Variable, and setting the same output in right and left

4.5.3 SetSysUSBMode

- a) Explanation of Commands : Set USB Mode. This FS Command only works normally with foreign firmwares.
- b) Input Value : '0 ~ 1'
 - '0' : MSC
 - '1' : MTP
- c) Return Value : '-1 ~ 1'
 - '-1' : When Input Value exceeds the limit
 - '0' : When 'MSC' is set normally
 - '1' : When 'MTP' is set normally
- d) Example

```
var value:Number = ext_fscommand2("SetSysUSBMode", "0");
```
- e) Example Explanation : Return '0' to 'value' variable, and when USB is connected 'MSC' is activated.

4.5.4 SetSysDefault

- a) Explanation of Commands : Restores system default settings. However, certain settings such as language settings are not initialized. This FS Command does not save settings values depending on device operations.
- b) Input Value : '0 ~ 1'

- '0' : No action
 - '1' : Load the initial Setting
- c) Return Value : '-1 ~ 1'
- '-1' : When Input Value exceeds the limit
 - '0 ~ 1' : Return the input value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetSysDefault", "1");
```

e) Example Explanation : Return '1' to 'value' variable, and loading the initial setting

4.5.5 SetSysClearHDS

a) Explanation of Commands : Initialize MS DRM information with the FS Command which is only used with foreign firmwares. This FS Command doesn't save the setting value with actions

b) Input Value : '0 ~ 1'

- '0' : No action
- '1' : Initialize MS DRM

c) Return Value : '-1 ~ 1'

- '-1' : When Input Value exceeds the limit
- '0 ~ 1' : Return the Input Value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetSysClearHDS", "1");
```

e) Example Explanation : Return '1' to 'value' variable, and initializing MS DRM information

4.5.6 SetSysSleepMode

a) Explanation of Commands: Set Sleep Mode

b) Input Value : '0 ~ 1'

- '0' : Enter into 'Power off' state when the device is turned off
- '1' : Enter into 'Sleep' state when the device is turned off

c) Return Value : '-1 ~ 1'

- '-1' : When Input Value exceeds the limit
- '0 ~ 1' : Return the Input value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetSysSleepMode", "0");
```

e) Example Explanation : Return '0' to 'value' variable, and entering into 'Power Off' state when the device is turned off.

4.5.7 SetSysBeepMode

a) Explanation of Commands : Set Beep Mode. Beep Sound occurs once when touch action is triggered

b) Input Value : '0 ~ 4'

- '0' : Off
- '1' : Waterdrop
- '2' : Balloon
- '3' : Bell 1
- '4' : Bell 2

c) Return Value : '-1 ~ 4'

- '-1' : When Input Value exceeds the limit

- '0 ~ 4' : Return the input value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetSysBeepMode", "2");
```

e) Example Explanation : Return '2' to 'value' variable, and generating 'Balloon' sound when the touch panel is pressed.

4.5.8 SetSysOffFlag

a) Explanation of Commands : Turn off the device by FS Command. This FS Command turns off the device when the Frame is over by Asynchronous FS Command unless there's no other input values triggering actions.

b) Input Value : '0 ~ 2'

- '0' : No action
- '1' : Sleep
- '2' : Power Off

c) Return Value : '-1 ~ 2'

- '-1' : When Input Value exceeds the limit
- '0 ~ 2' : Return the input value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetSysOffFlag", "1");
```

e) Example Explanation : Return '1' to 'value' variable, and switch the device to 'Sleep' state when the frame called by Flash FS Command finishes..

4.6 Music Mode Set Parameter FS Command

Explaining the FS Command that is used for setting Music Value and Music GUI Value

4.6.1 SetAudSkipLength

a) Explanation of Commands : Setting 'Skip Length' (FF/REW Short Key)

b) Input Value : '0 ~ 7'

- '0' : Track
- '1' : 2 sec
- '2' : 3 sec
- '3' : 4 sec
- '4' : 5 sec
- '5' : 10 sec
- '6' : 20 sec
- '7' : 30 sec

c) Return Value : '-1 ~ 7'

- '-1' : When Input Value exceeds the limit
- '0 ~ 7' : Return the input value when it works normally

d) Example

```
var value:Number = ext_fscommand2("SetAudSkipLength", "5");
```

e) Example Explanation : Return '5' to 'value' variable, and setting the skip length as '10' sec

4.6.2 SetAudScanSpeed

a) Explanation of Commands : Return Scan Speed (FF/REW Long Key)

b) Input Value : '0 ~ 4'

- '0' : 3 sec
- '1' : 5 sec
- '2' : 10 sec

- '3' : 20 sec
 - '4' : 30 sec
- c) Return Value : '-1 ~ 4'
- '-1' : When Input Value exceeds the limit
 - '0 ~ 4' : Return the input value when it works normally
- d) Example
- ```
var value:Number = ext_fscommand2("SetAudScanSpeed", "0");
```
- e) Example Explanation : Return '0' to 'value' variable and setting the scan speed as '3 sec'

#### 4.6.3 SetAudResume

- a) Explanation of Commands : Set Audio Resume.
- b) Input Value : '0 ~ 1'
- '0' : Off
  - '1' : On
- c) Return Value : '-1 ~ 1'
- '-1' : When Input Value exceeds the limit
  - '0 ~ 1' : Return the input value when it works normally
- d) Example
- ```
var value:Number = ext_fscommand2("SetAudResume", "1");
```
- e) Example Explanation : Return '1' to 'value' variable and playing music from the latest music file after starting Music mode.

4.6.4 SetAudPSpeed

- a) Explanation of Commands : Set Audio Play Speed
- b) Input Value : '0 ~ 10'
- '0' : 50%
 - '1' : 60%
 - '2' : 70%
 - '3' : 80%
 - '4' : 90%
 - '5' : 100%
 - '6' : 110%
 - '7' : 120%
 - '8' : 130%
 - '9' : 140%
 - '10' : 150%
- c) Return Value : '-1 ~ 10'
- '-1' : When Input Value exceeds the limit
 - '0 ~ 10' : Return the input value when it works normally
- d) Example
- ```
var value:Number = ext_fscommand2("SetAudPSpeed", "5");
```
- e) Example Explanation : Return '5' to 'value' variable and playing music '100%' speed

#### 4.6.5 SetAudFadeln

- a) Explanation of Commands : Set Audio Fade In
- b) Input Value : '0 ~ 5'
- '0' : Off
  - '1' : 1 sec
  - '2' : 2 sec
  - '3' : 3 sec

- '4' : 4 sec
- '5' : 5 sec
- c) Return Value : '-1 ~ 5'
  - '-1' : When Input Value exceeds the limit
  - '0 ~ 5' : Return the input value when it works normally
- d) Example
 

```
var value:Number = ext_fscommand2("SetAudFadeIn", "0");
```
- e) Example Explanation : Return '0' to 'value' variable and setting 'Fade In' off while playing music

#### 4.6.6 SetAudRepeat

- a) Explanation of Commands : Setting Audio Repeat
- b) Input Value : '0 ~ 1'
  - '0' : Off
  - '1' : On
- c) Return Value : '-1 ~ 1'
  - '-1' : When Input Value exceeds the limit
  - '0 ~ 1' : Return the input value when it works normally
- d) Example
 

```
var value:Number = ext_fscommand2("SetAudRepeat", "1");
```
- e) Example Explanation : Return '1' to 'value' variable and playing the play list repeatedly

#### 4.6.7 SetAudShuffle

- a) Explanation of Commands : Sets shuffle. When there are ten songs on the playlist, shuffle plays them back in a random sequence such as "5, 1, 3, 8, 9, 0, 2, 7, 4, 6". When the shuffle function turns 'On', the song that is currently selected or playing is assigned to index 0. Also, when the shuffle turns 'On' differently than the previous model, the playlist that is mixed at this time is saved to the device. The playlist is re-shuffled in the following cases.
  - When shuffle mode is on.
  - create new play list by selecting music files in browser.
  - Playback the last song by click "Rew" button while listening the first music in the play list.
  - Playback the Fist song by click "FF" button while listening the last music in the play list.
  - When returning to the first song as the last file play finishes.
  - When the quantity of the play list is changed by changing playback boundary
  - When the quantity of the play list is changed by deleting files.
  - When the quantity of the play list is changed by delete files while connected to PC.
- b) Input Value : '0 ~ 1'
  - '0' : Off
  - '1' : On
- c) Return Value : '-1 ~ 1'
  - '-1' : If it is not working properly because Input Value is exceeded the limits.
  - '0 ~ 1' : return Input Value if it is working properly.
- d) Example
 

```
var value:Number = ext_fscommand2("SetAudShuffle", "1");
```
- e) Example Explanation: Return '1' to 'Value' Variable and play back shuffle in play list.

#### 4.6.8 SetAudBoundary

- a) Explanation of Commands : Sets the playback boundary. While the playback boundary operates according to the input value listed below when the file is selected from the folder browser, 'All Files' operates in the same way as 'Folder' when the file is selected in the tag browser. Because a tag is a virtual DB, various tags can be selected even though the tag browser is the same file.
- b) Input Value : '0 ~ 2'
- '0' : All
  - '1' : One
  - '2' : Folder
- c) Return Value : '-1 ~ 2'
- '-1' : if it is not working properly by Input Value is exceeded limits.
  - '0 ~ 2' : return 'input value' when it is working properly.
- d) Example
- ```
var value:Number = ext_fscommand2("SetAudBoundary", "0");
```
- e) Example Explanation : return '0' to 'value' variable and set as play all boundary when a file is selected to play in current folder.

4.6.9 SetAudAlbumArtIndex

- a) Explanation of Commands : Sets the album artwork to be loaded when 'loadMovie' is executed. Certain sound sources can have multiple album artworks in a single file. To support such functions, the S9 can call up to six album artworks. If 'loadMovie' is executed with higher number of album artworks than what the sound source contains, the album artwork that belongs to index 0 is loaded.
- b) Input Value : '0 ~ 5'
- '0 ~ 5' : index of loadable album art.
- c) Return Value : '-1 ~ 5'
- '-1' : if it is not able to work because Input Value is exceeded limits.
 - '0 ~ 5' : return input value if it is working correctly.
- d) Example
- ```
var value:Number = ext_fscommand2("SetAudAlbumArtIndex", "1");
```
- e) Example Explanation : Returns '1' to the 'value' variable and loads the index '1' album artwork when loadMovie is executed.

#### 4.6.10 SetAudLDBConfig

- a) Explanation of Commands : Sets lyrics (LDB). Since lyrics are controlled directly by the Low Level class to improve the processing speed, the GUI area that will accommodate lyrics must be secured in the ActionScript. The settings value is a string that uses the '|' separator. LDB 1.0 uses an external program, the 'AUDIO LDB Manager', to edit lyrics, which is not supported outside of a lyrics window whose size is fixed. LDB 2.0, on the other hand, edits the lyrics internally, which makes it possible to control the size of the lyrics window in the ActionScript.
- b) Input Value : String
- First Value : On / Off
  - Second Value : X coordinate
  - Third Value : Y coordinate
  - Fourth Value : Width(Fixed at '200' in LDB 1.0 regardless of input value)
  - Fifth Value : Height(Fixed at '60' in LDB 1.0 regardless of input value)
  - Sixth Value : Number of lines (Fixed at '3' in LDB 1.0 regardless of input value)

- Seventh Value : Font size (Fixed at '16' in LDB 1.0 regardless of input value)
  - Eighth Value : 24bit(RGB) Font Color
  - Ninth Value : 24 bit (RGB) lyrics box color used in LDB 2.0. (Not used for LDB 1.0)
- c) Return Value : '-1 ~ 1'
- '-1' : if it is not working properly cause of Input Value is exceeded the limits.
  - '0 ~ 1': return On or Off according to the first input value.
- d) Example
- ```
var value:Number = ext_fscommand2("SetAudLDBConfig", "1|39|300|200|60|3|16|16777215|16711680");
```
- e) Example Explanation : if it is playing a file with lyric that is made by LDB 2.0, return '1' to 'value' variable and set lyrics as below.
- First value : On
 - Second value : 39 Pixel
 - Third value : 300 Pixel
 - Fourth value : 200 Pixel
 - Fifth value : 60 Pixel
 - Sixth value: 3 row
 - Seventh value : 16 Pixel
 - Eighth value : 16777215(0xFFFFFFFF, white)
 - Ninth value : 16711680(0xFF0000, red)

4.7 Video Mode Set Parameter FS Command

Explaining FS Command relating to the value in both Video and Music GUI

4.7.1 SetVidReOpen

- a) Action Explanation : Reopen the files by force and initialize. When the start mode is Bookmark or Favoring with checking GetVidStartMode, this command should always be called first.
- b) Input Value : None
- c) Output Value :
- '-1' : When the fails to work
 - '0' : When it works normally.
- d) Example
- Please refer to GetVidStartMode example

4.7.2 SetVidATime

- a) Action Explanation : In Video A-B mode, sets A time with millisecond
- b) Input Value : None
- c) Output Value :
- '-1' : When the command fails to work normally.
 - '0' : When it works normally.
- d) Example
- ```
/* Sample of SetVidATime
: Example for A-B mode setting */
ext_fscommand2("SetVidABOff", 0); // A-B Mode off
ext_fscommand2("SetVidATime", 10*1000); // Setting 10 sec as A time
ext_fscommand2("SetVidBTime", 20*1000); // Setting 20 sec B time
ext_fscommand2("SetVidABOff", 1); // A-B Mode on
```

### 4.7.3 SetVidBTime

- a) Action Explanation : Setting B time with millisecond in Video A-B Mode
- b) Input Value : None

- c) Output Value :
- '-1' : When the command fails to work normally.
  - '0' : When it works normally.

d) Example

Please refer to SetVidATime Example

#### 4.7.4 SetVidABOff

- a) Action Explanation : A-B Mode on/off

b) Input Value :

- '0' : Video A-B Mode off
- '1' : Video A-B Mode on.

c) Output Value :

- '-1' : When the command fails to work normally.
- '0' : When it works normally.

d) Example

Please refer to SetVidATime Example.

#### 4.7.5 SetVidSkipLength

- a) Action Explanation : Setting Video Skip Length. Skip length means key frame's number. Therefore 0 indicates skipping 1 frame and 7 indicates skipping 8 frames.

b) Input Value :

- '0~7' : The wanted skip frame + 1

c) Output Value :

- '-1' : When the command fails to work normally.
- '0' : The set value.

d) Example

```
ext_fscommand2("SetVidSkipLength", 2); // Skipping 3 frames
```

#### 4.7.6 SetVidResume

- a) Action Explanation : Setting-> Video -> Setting the resume value

b) Input Value :

- '0' : Inactivating resume function
- '1' : Activating resume function

c) Output Value :

- '-1' : When the command fails to work normally.
- '0 ~1' : Set Value

d) Example

```
ext_fscommand2("SetVidResume", 1); // Change the resume value to ON
```

#### 4.7.7 SetVidPSpeed

- a) Action Explanation : Setting->Video -> Setting Play Speed

b) Input Value :

- '0~10' : Indicating 50% ~150% of the original speed

c) Output Value :

- '-1' : When the command fails to work normally.
- '0 ~10' : Set Value

d) Example

Please refer to GetVidPSpeed Example

#### 4.7.8 SetVidCaption

- a) Action Explanation : Setting Device -> Video -> Setting the value of Caption
- b) Input Value :
  - '0' : Inactivate the caption function
  - '1' : Activate the caption function
- c) Output Value
  - '-1' : When the command fails to work normally.
  - '0 ~1' : Set Value.
- d) Example

```
ext_fscommand2("SetVidCaption", 1); // Activating the caption setting
ext_fscommand2("SetVidCaption", 0); // Inactivating the caption setting
```

#### 4.7.9 SetVidRepeat

- a) Action Explanation : Setting Video repeat
- b) Input Value : '0 ~ 1'
  - '0' : Off
  - '1' : On
- c) Output Value : '-1 ~ 1'
  - '-1' : When it exceeds the limit of Input Value
  - '0 ~ 1' : Return the input Value when it works normally
- d) Example

```
var value:Number = ext_fscommand2("SetVidRepeat", "1");
```

#### 4.7.10 SetVidBoundary

- a) Action Explanation : Setting Video Play Boundary. Play Boundary works with the below input value when the file is selected in Folder Browser, but when the file is selected in Tag Browser all files work the same with 'Folder'. It is because Tag is virtue DB so that Tag Browser is the same file, but it can also be selected.
- b) Input Value : '0 ~ 2'
  - '0' : All files
  - '1' : One file
  - '2' : Folder
- c) Output Value : '-1 ~ 2'
  - '-1' : When Input Value exceeds the limit
  - '0 ~ 2' : Return Input Value when it works normally
- d) Example

```
var value:Number = ext_fscommand2("SetVidBoundary", "0");
```

#### 4.7.11 SetVidColorKey

- a) Action Explanation : Setting Color key's value on Video UI display
- b) Input Value : String
  - First Value : Chorma(Color)Key 24bit RGB Value
  - Second Value' : Deactivation/Activation (0/1)
- c) Output Value :
  - '-1' : When the command fails to work normally.

- '1' : When it works properly

d) Example

```
// Setting Color key's value as 0x010203(0xRRGGBB).
ext_fscommand2("SetVidColorKey", 0x010203, 1);
```

#### 4.7.12 SetVidAlpha

a) Action Explanation : Setting Alpha Value on Video UI display

b) Input Value : String

- First Value : 4bit Alpha (0~15);
- 'Second Value : Deactivation/Activation(0/1)

c) Output Value :

- '-1' : When the command fails to work normally.
- '1' : When it works normally

d) Example

```
// Setting Alpha Value as 7 (46%).
ext_fscommand2("SetVidAlpha", 7, 1);
```

#### 4.7.13 SetVidStartMode

a) Action Explanation : Setting Video Start Mode before opening the file

b) Input Value :

- 0: Basic Mode
- 1: Resume Mode

c) Output Value :

- '-1' : When the command fails to work normally.
- '0~1': Return Input Value

d) Example

Please refer to GetVidStartMode Example

#### 4.7.14 SetVidSCR

a) Action Explanation : Setting Video Screen with Video Play Mode

b) Input Value : String

- First Value : x coordinates
- Second Value : y coordinates
- Third Value : The width of a screen
- Fourth Value : The height of a screen
- Fifth Value : Screen scale (0/1/2/3:90 degree/180 degree/270 degree / 360 degree)
- Sixth Value : Scale Mode (0/1/2: Filling a screen in line with Auto/Ratio)
- Seventh Value : Setting the default screen on/off (1/0)

c) Output Value :

- '-1' : When the command fails to work normally.
- '1' : When it works normally

d) Example

```
ext_fscommand2("SetVidSCR", 0, 0, 272, 480, 0, 2, 1);
```

#### 4.7.15 SetVidCaptionSyncOffset

a) Action Explanation : Setting Sync Offset with a Millisecond

b) Input Value :

- offset: Setting Offset with a Millisecond unit

c) Output Value :

- '-1' : When the command fails to work normally.

- '0' : When it works normally

d) Example

```
// Shifting the caption to two seconds later
ext_fscommand2("SetVidCaptionSyncOffset", 2*1000);
// Shifting the caption to two seconds earlier
ext_fscommand2("SetVidCaptionSyncOffset", -2*1000);
```

#### 4.7.16 SetVidTVOut

a) Action Explanation : Setting TV-out Mode

b) Input Value :

- It should always be set as 1

c) Output Value :

- '-1' : When the command fails to work normally.

- '0' : When it works normally

d) Example

```
ext_fscommand2("SetVidTVOut", 1); // Being Changed to TV-out Mode
```

#### 4.7.17 SetVidUIEnable

a) Action Explanation : Setting Video UI on/off

b) Input Value :

- 0: on

- 1: off

c) Output Value :

- '-1' : When the command fails to work normally.

- '0' : When it works normally

d) Example

```
ext_fscommand2("SetVidUIEnable", 0); // UI off
ext_fscommand2("SetVidUIEnable", 1); // UI on
```

#### 4.7.18 SetVidCaptionEnable

a) Action Explanation : Setting Video Caption on/off

b) Input Value :

- 0: on

- 1: off

c) Output Value :

- '-1' : When the command fails to work normally.

- '0' : When it works normally

d) Example

Please refer to GetVidCaption Example

### 4.8 Record Mode Set Parameter FS Command

Explain FS command that set value of recording and value of recording GUI.

#### 4.8.1 SetRecLineInbps

a) Explanation of Commands : set line in recording quality.

b) Input Value : '0 ~ 5'

- '0' : 32 kbps

- '1' : 64 kbps

- '2' : 80 kbps

- '3' : 96 kbps
- '4' : 128 kbps
- '5' : 256 kbps
- c) Return Value : '-1 ~ 5'
  - '-1' : if it is not working properly causing of Input Value is exceeded the limits.
  - '0 ~ 5' : return the in put value when it is working right.
- d) Example
 

```
var value:Number = ext_fscommand2("SetRecLineInbps", "3");
```
- e) Example Explanation : return '3' to 'value' variable and record line in to 96 kbps.

#### 4.8.2 SetRecVoicebps

- a) Explanation of Commands : Set quality of voice recording.
- b) Input Value : '0 ~ 5'
  - '0' : 32 kbps
  - '1' : 64 kbps
  - '2' : 80 kbps
  - '3' : 96 kbps
  - '4' : 128 kbps
  - '5' : 256 kbps
- c) Return Value : '-1 ~ 5'
  - '-1' : if it is not working properly causing of Input Value is exceeded the limits.
  - '0 ~ 5' : return the in put value if it is working correct.
- d) Example
 

```
var value:Number = ext_fscommand2("SetRecVoicebps", "4");
```
- e) Example Explanation : return '4' to 'value' variable and record voice to 128 kbps.

#### 4.8.3 SetRecFMbps

- a) Explanation of Commands : set quality of FM radio recoding.
- b) Input Value : '0 ~ 5'
  - '0' : 32 kbps
  - '1' : 64 kbps
  - '2' : 80 kbps
  - '3' : 96 kbps
  - '4' : 128 kbps
  - '5' : 256 kbps
- c) Return Value : '-1 ~ 5'
  - '-1' : if it is not working properly causing of Input Value is exceeded the limits.
  - '0 ~ 5' : if it is not working properly causing of Input Value is exceeded the limits.
- d) Example
 

```
var value:Number = ext_fscommand2("SetRecFMbps", "5");
```
- e) Example Explanation: return '3' to 'value' variable and record FM radio to 256 kbps.

#### 4.8.4 SetRecLineInVolume

- a) Explanation of Commands : set recording volume of line in recording.
- b) Input Value : '0 ~ 4'
  - '0' : Volume 1
  - '1' : Volume 2
  - '2' : Volume 3
  - '3' : Volume 4

- '4' : Volume 5
- c) Return Value : '-1 ~ 4'
  - '-1' : if it is not working properly causing of Input Value is exceeded the limits.
  - '0 ~ 4' : return the in put value when it is working right.
- d) Example
 

```
var value:Number = ext_fscommand2("SetRecLineInVolume", "3");
```
- e) Example Explanation : return '3' to 'value' variable and set volume as 4 when it is line in recording.

#### 4.8.5 SetRecVoiceVolume

- a) Explanation of Commands : set recording volume of voice recording.
- b) Input Value : '0 ~ 4'
  - '0' : Volume 1
  - '1' : Volume 2
  - '2' : Volume 3
  - '3' : Volume 4
  - '4' : Volume 5
- c) Return Value : '-1 ~ 4'
  - '-1' : if it is not working properly causing of Input Value is exceeded the limits.
  - '0 ~ 4' : return the in put value if it is working right.
- d) Example
 

```
var value:Number = ext_fscommand2("SetRecVoiceVolume", "4");
```
- e) Example Explanation : return '4' to 'value' variable and set volume as 4 if it is operate voice recording.

#### 4.8.6 SetRecAutoSync

- a) Explanation of Commands : Sets the automatic synchronization function used during line-in recording.  
If there is no recording input for a certain length of time during line-in recording, automatic synchronization stops the current recording and continues after generating a new file.
- b) Input Value : '0 ~ 1'
  - '0' : Off
  - '1' : On
- c) Return Value : '-1 ~ 1'
  - '-1' : if it is not working properly causing of Input Value is exceeded the limits.
  - '0 ~ 1' : return the in put value when it is working right.
- d) Example
 

```
var value:Number = ext_fscommand2("SetRecAutoSync", "1");
```
- e) Example Explanation: return '1' to 'value' variable and operate auto sync function in line in recording.

#### 4.8.7 SetRecMode

- a) Explanation of Commands : set recording mode.
- b) Input Value : '0 ~ 1'
  - '0' : Line-in Recording
  - '1' : MIC Recording
- c) Return Value : '-1 ~ 1'
  - '-1' : if it is not working properly because Input Value is exceeded the limits.
  - '0 ~ 1' : return the in put value when it is working right.
- d) Example
 

```
var value:Number = ext_fscommand2("SetRecMode", "1");
```
- e) Example Explanation : return '1' to 'value' variable and set voice recording mode..

## 4.9 FM Radio Mode Set Parameter FS Command

Explanation of FS Commands which set values related to FM radio and values required for FM radio GUI's. During an FM radio auto scan, Set FS Command does not operate but returns '-1'.

### 4.9.1 SetFmrStereo

- a) Explanation of Commands : setting FM Radio Stereo
- b) Input Value : '0 ~ 1'
  - '0' : Mono
  - '1' : Stereo
- c) Return Value : '-1 ~ 1'
  - '-1' : when it is not working properly causing of Input Value is exceeded the limits.
  - '0 ~ 1' : return the in put value when it is working right.
- d) Example

```
var value:Number = ext_fscommand2("SetFmrStereo", "1");
```
- e) Example Explanation : return '1' to 'value' variable and set voice recording mode.

### 4.9.2 SetFmrAutoScan

- a) Explanation of Commands : operate auto scan reception of FM radio. Auto scan works through between the minimum and the maximum frequency and saves valid frequencies into radio presets.
- b) Input Value : None
- c) Return Value : '76000, '87500'
  - '76000' : when region is set as Japan for FM radio.
  - '87500' : when region is set as other countries then Japan for FM radio.
- d) Example

```
var value:Number = ext_fscommand2("SetFmrAutoScan");
```

### 4.9.3 SetFmrRegion

- a) Explanation of Commands : Set region of FM radio.
- b) Input Value : '0 ~ 3'
  - '0' : China & Europe
  - '1' : Japan
  - '2' : Korea
  - '3' : US
- c) Return Value : '-1 ~ 3'
  - '-1' : if it is not working properly because Input Value is exceeded the limits.
  - '0 ~ 3' : return the input value, if it is working right.
- d) Example

```
var value:Number = ext_fscommand2("SetFmrRegion", "2");
```
- e) Example Explanation : return '3' to 'value' variable and set region of FM radio to 'Korea'

### 4.9.4 SetFmrCurrentFreq

- a) Explanation of Commands : Sets the FM radio frequency. If FM is in operation, the frequency changes and the unit plays. If not, only the frequency changes.
- b) Input Value : '76000 ~ 118000' / '87500 ~ 118000'((unit) kHz)
  - '76000 ~ 118000' : when region is set as Japan for FM radio
  - '87500 ~ 118000' : when region is set as other countries then Japan for FM radio.
- c) Return Value : '-1', '76000 ~ 118000'
  - '-1' : when it is not working properly causing of Input Value is exceeded the limits.

- '76000 ~ 118000' : return the input value when it is working right.

d) Example

```
var value:Number = ext_fscommand2("SetFmrCurrentFreq", "91900");
```

e) Example Explanation : '91900' is returned to the 'value' variable and the FM radio is tuned to 91.9 Mhz.

#### 4.9.5 SetFmrPresetOn

a) Explanation of Commands : set the preset of FM radio.

b) Input Value : '0 ~ 1'

- '0' : Off

- '1' : On

c) Return Value : '-1 ~ 1'

- '-1' : when it is not working properly causing of Input Value is exceeded the limits.

- '0 ~ 1' : return the input value when it is working right.

d) Example

```
var value:Number = ext_fscommand2("SetFmrPresetOn", "1");
```

e) Example Explanation : return '1' to value of variable and FM radio is operating by preset mode.

#### 4.9.6 SetFmrPresetCurNum

a) Explanation of Commands : Sets the FM radio preset index. As with the 'SetFmrCurrentFreq' FS Command, if FM is in operation, the frequency changes to the preset frequency and the unit plays. If not, the frequency only changes to the preset frequency. If the applicable preset frequency is not valid, the frequency does not change.

b) Input Value : '0 ~ 23'

- '0 ~ 23' : index of preset

c) Return Value : '0 ~ 23'

d) Example

```
var value:Number = ext_fscommand2("SetFmrPresetCurNum", "3");
```

e) Example Explanation : '3' is returned to the 'value' variable and, if the frequency of the FM radio preset index '3' is 91.9 MHz, the current FM radio is changed to 91.9 MHz.

#### 4.9.7 SetFmrPresetFreq

a) Explanation of Commands : Changes the frequency of the applicable FM radio preset index to the frequency input.

b) Input Value : '0 ~ 23', '0, 76000 ~ 118000' / '0, 87500 ~ 118000'((unit) kHz)

First Input Value

- '0 ~ 23' : FM Radio Freeset Index

Second Input Value

- '0, 76000 ~ 118000' : when region is set as Japan for FM radio

- '0, 87500 ~ 118000' : when region is set as other countries then Japan for FM radio.

c) Return Value : '-1', '0, 76000 ~ 118000'

- '-1' : when it is not working properly causing of Input Value is exceeded the limits.

- '0, 76000 ~ 118000' : return the input value when it is working right.

d) Example

```
var value:Number = ext_fscommand2("SetFmrPresetCurNum", "4", "91900");
```

```
var value2 :Number = ext_fscommand2("SetFmrPresetCurNum", "5", "0");
```

e) Example Explanation : In the first example, '91900' is returned to the 'value' variable and the frequency of

the FM radio preset index '4' is changed to 91.9 Mhz. In the second example, '0' is returned to the 'value2' variable and the frequency of the FM radio preset index '5' is changed to 0 Mhz.

#### 4.10 Mobile TV(DMB/DAB) Mode Set Parameter FS Command

Mobile TV Mode Set Parameter FS Command will be revealed soon.

#### 4.11 Etc. Set Parameter FS Command

Explain setting FS Command of other Parameter.

##### 4.11.1 SetEtcUIConfig

a) Explanation of Commands : This FS Command saves GUI setting variables in each Flash as characters.

For more detailed information, please refer to the 'GetEtcUIConfig'

FS Command section.

b) Input Value : '0 ~ 15', string

First Input Value

- '0' : Music
- '1' : Video
- '2' : Radio
- '3' : Record
- '4' : MobileTV
- '5' : Flash
- '6' : Document
- '7' : Picture
- '8' : Dictionary
- '9' : Util
- '10' : Etc.
- '11' : Mainmenu
- '12' : Mainmenu2
- '13' : Mainmenu3
- '14' : Setting
- '15' : Browser

Second Input Value

- An array of strings needs to be saved

c) Return Value : '-1' '1'

- '-1' : when it is not working properly causing of Input Value is exceeded the limits or text file is bigger than 128 byte

- '1' : when it is working properly.

d) Example

```
var musicData:String = "3|20|400|30";
```

```
var value:Number = ext_fscommand2("SetEtcUIConfig", "0", musicData);
```

e) Example Explanation

'1' is returned to the 'value' variable and saves the "3|20|400|30" string to the Music UI storage space.

##### 4.11.2 SetEtcSettingInit

a) Explanation of Commands : initialize setting menu. Must be called upon entering the settings menu the first time.

b) Input Value : None

c) Return Value : '0 ~'

- '0 ~' : return the number of list of 'Step 0'.

d) Example

```
var value:Number = ext_fscommand2("SetEtcSettingInit");
```

#### 4.11.3 SetEtcSettingValue

a) Explanation of Commands : apply the setting value in setting menu 'step 2'.

b) Input Value : setting value of current menu.

c) Return Value : '-1 ~'

- '-1' : If step of menu is set as '2' or Input Value is exceeded the limits.

- '0 ~' : return the input value if it is working right.

d) Example

```
var value:Number = ext_fscommand2("SetEtcSettingValue", "1");
```

e) Example Explanation

After selecting 'Settings (Step 0) -> Display (Step 1) -> Fonts (Step 2)' and calling the FS Command as in the example saves '1' to the 'value' variable and the font is set to 'user fonts'.

#### 4.11.4 SetEtcSettingNext

a) Explanation of Commands : move to next setting menu.

b) Input Value : index of changed menu list.

c) Return Value : '-1 \* Input Value', '0 ~'

- '-1 \* Input Value' : Converts the index input into a negative number upon entering the information menu and returns it

- '0 ~' : return the number of new menu list after changing menu.

d) Example

```
var value:Number = ext_fscommand2("SetEtcSettingValue", "1");
```

e) Example Explanation

Calling the FS Command as in the example in 'Settings (Step 0)' saves '7' to the 'value' variable and the settings menu changes to 'Display (Step 1)'.

#### 4.11.5 SetEtcSettingPrev

a) Explanation of Commands : move to above setting menu.

b) Input Value : None

c) Return Value : '0 ~'

- '0 ~' : return the new menu list number after changing.

d) Example

```
var value:Number = ext_fscommand2("SetEtcSettingPrev");
```

e) Example Explanation

Calling the FS Command as in the example in 'Display (Step 1)' saves '8' to the 'value' variable and the settings menu changes to 'Settings (Step 0)'.

#### 4.11.6 SetEtcCurPLIndex

a) Explanation of Commands : An FS Command used in the Music, Video, and Picture Mode, this command sets the playback index on the playlists (Music, Video) or the picture lists (Picture). The index value ranges from '0' to 'Total number of playlists -1'. To ensure proper operation in the Music or Video Mode, the playback must stop first before changing the playback index. Also, if loadMovie is executed on the pictures in the ActionScript as in the picture slide show, this FS Command must be used to convey to the Low Level class the index of the files on which loadMovie is executed. When the browser is run afterwards, however, the

browser settings are set to the most recently viewed file.

b) Input Value : '0 ~'

- '0 ~' : setting index in applicable list.

c) Return Value : '-1 ~'

- '-1' : when it is not working properly causing of Input Value is exceeded the limits or it is not in Music, Video, or Picture mode.

- '0 ~' : return the in put value when it is working right.

d) Example

```
var value:Number = ext_fscommand2("SetEtcCurPLIndex", "2");
```

e) Example Explanation

With the Music Mode stopped and with the total of 10 playlists ('GetEtcTotalPLNum'), calling the FS Command as in the example saves '2' to the 'value' variable and the file that belongs to the '2' index is selected.

#### 4.11.7 SetEtcLCDTimerOff

a) Explanation of Commands : Controls the auto screen off timer. The Low Level class is set to turn off the screen automatically according to the 'SetDisLCDAutoOff' FS Command setting. There are times, however, when the screen must be left turned on, as in viewing a document or a picture. In such cases, the auto screen off timer can be set so that it is not used, leaving the screen on. Also, if this FS Command is used to shut off the timer, the auto stand-by timer set by 'SetTimeAutoOff' does not work either, leaving the unit on. Leaving the screen always on requires great power consumption, so please be cautious this with FS Command.

b) Input Value : '0 ~ 1'

- '0' : use auto power off function.

- '1' : not using auto power off function.

c) Return Value : '-1 ~ 1'

- '-1' : when it is not working properly causing of Input Value is exceeded the limits.

- '0 ~ 1' : return the in put value when it is working right.

d) Example

```
var value:Number = ext_fscommand2("SetEtcLCDTimerOff", "1");
```

e) Example Explanation

Save '1' to value variable and auto off timer is off. After this UI it must re activate the auto off timer.

#### 4.11.8 SetEtcWallpaper

a) Explanation of Commands : Sets the current background as wallpaper. When wallpaper is set normally, input value will be set as '1' in the firmware. The selected wallpaper can be called if perform loadMovie the file named 'WALLPAPER.BNG'. However, 'SetDisWallpaper' FS Command must be set as '1' to normally loadMovie the wallpaper.

b) Input Value : None

c) Return Value : '-1 ~ 0'

- '-1' : Not processed due to firmware issue

- '0' : when processed normally

d) Example

```
var value:Number = ext_fscommand2("SetEtcWallpaper");
```

e) Example explanation :

'0' will be saved under 'value' string, and the current background will be set as wallpaper. This FS

Command is called while Frame being processed, so screen processing must be finished, and called at the next frame.

## 5 Etc. FS Command

### 5.1 Browser FS Command

Explains browser related FS Command.

#### 5.1.1 EtcBrwSetInitialization

a) Explanation of Action : Initialize Browser. FS Command related to Browser guarantees normal function after initializing it. Browser is categorized by 'General', 'Music Album', 'Photo', 'Book Mark', and 'Favorite'. And Music, Video each has two types of play lists. One is the list made by selecting the files from 'General' browser, and the other is the list made by selecting the files from 'Favorite' browser. Therefore one shall need to check the list first if it is made from 'General' or 'Favorite' before initializing browser. When the file is selected in 'Book Mark' browser, the play list is made based on the location of the selected file, so the list is categorized as 'General' browser without any particular check. 'Music Album' browser consists of 'Album' Tag and 'Title' Tag detached from 'General Browser'. Accordingly, it works only with Music Mode, and the initializing shall be located in 'Album' Tag and be only moved to 'Album' Tag and 'Title' Tag.

b) Input Value : String

- 'Default' : Be initialized to 'General' Browser.
- 'Album' : Be initialized to 'Music Album' Browser.
- 'Picture' : Be initialized to 'Picture' Browser. 'Picture' Browser only shows the list related to the picture files, differently from 'General' Browser.
- 'Bookmark' : Be initialized to 'Book Mark'. 'Book Mark' has the one stage as a virtual list. 'Book Mark' function only works on Music, Video, Document Mode, since it's only supported with Music, Video, Document Mode.
- 'Favorite' : Be initialized to 'Favorite'. 'Favorite' has the one stage as a virtual list. 'Favorite' function only works on Music, Video, Document Mode, since it's only supported with Music, Video, Document Mode.

c) Output Value : '-1 ~'

- '-1' : When the string is wrong or being inappropriately initialized to 'Book Mark' or 'Favorite' Browser.
- '0 ~' : Returning the total number of Browser list when it works normally

d) Example

```
var browserMode:String = "Default";
var value:Number = ext_fscommand2("EtcBrwSetInitialization", browserMode);
```

e) Example Explanation

Be initialized to 'General' Browser, and the number of browser lists is saved in 'value' variable.

#### 5.1.2 EtcBrwSetNextStage

a) Explanation of Commands : Changes the browser stage into a sub-stage and updates the browser list.

This FS Command initiates the following browser actions:

- When input index is 'folder' : moves to selected stage by folder
- When input index is 'tag' : moves to selected stage by tag
- When input index is 'file' : selected file will run, and the type of file will be saved as string under

'variable'. Run the relevant Flash GUI file according to the string saved under 'variable' in ActionScript.

b) Input Value : '0' ~ Current total number of browser list

'-1', String Type Variable Name

First Input Value

- '0' ~ ' Current total number of browser list

- '1' : Index of the browser list that can be moved or run

Second Input Value

- String Type Variable Name that file type will be saved when file is run.

\* The file that belongs to the selected browser index is music file, 'Music' string will be saved.

\* The file that belongs to the selected browser index is Video file, 'Video' string will be saved.

\* The file that belongs to the selected browser index is Picture file, 'Picture' string will be saved.

\* The file that belongs to the selected browser index is Text file, 'Text' string will be saved.

\* The file that belongs to the selected browser index is Flash file, 'Flash' string will be saved.

\* The selected browser index is not a file, empty string will be saved.

c) Return Value : '-2 ~'

- '-2' : When the file is run.

- '-1' : When Input Value is out of range, and cannot be processed normally.

- '0 ~' : When the stage is moved, the total number of the browser list in the new stage will be returned.

d) Example

```
var fileType:String;
```

```
var value:Number = ext_fscommand2("EtcBrwSetNextStage", "2", "fileType");
```

e) Example Explanation

If index '2' in the current browser list is a music file, '-2' is saved to the 'value' variable and the 'Music' string is saved to the 'fileType' variable. The example then creates a playlist with the selected music file and plays it.

### 5.1.3 EtcBrwSetPrevStage

a) Explanation of Commands : Changes the stage to an upper stage and updates the browser list.

If the current location is 'folder', it moves to the parent 'folder'. If the current location is 'tag', it moves to the upper 'tag'.

b) Input Value : None

c) Return Value : '-1 ~'

- '-1' : When the current location is the uppermost stage, 'bookmark', or the 'favorites' browser.

- '0 ~' : Stage When the stage is moved, the total number of the browser list in the new stage will be returned.

d) Example

```
var value:Number = ext_fscommand2("EtcBrwSetPrevStage");
```

### 5.1.4 EtcBrwGetListString

a) Explanation of Commands : This command saves the index input from the browser list to the variable.

In the case of the 'bookmark' browser, "list name|total playback time|time bookmarked" is saved to the variable.

b) Input Value : '0' ~ 'Total list of current browser - 1', String Type Variable Name

First Input Value

- '0' ~ 'Total list of current browser - 1' : Index where name will be induced.

Second Input Value

- String Type Variable Name that the names of list will be saved.

c) Return Value : '-1 ~ 9'

- '-1' : Input When Input Value is out of range, and cannot be processed normally.

- '0' : When the relevant index is 'Tag' from the browser list.
- '1' : When the relevant index is 'Folder' from the browser list.
- '2' : When the relevant index is 'Music' from the browser list.
- '3' : When the relevant index is 'Video' from the browser list.
- '4' : When the relevant index is 'Recorded file' from the browser list.
- '5' : When the relevant index is 'Flash file' from the browser list.
- '6' : When the relevant index is 'Text file' from the browser list.
- '7' : When the relevant index is 'Picture file' from the browser list.
- '8' : When the relevant index is 'Audible file' from the browser list.
- '9' : When there no file in the browser list, variable will be saved as 'No File'.

d) Example

```
var listName:String;
var value:Number = ext_fscommand2("EtcBrwGetListString", "2", "listName");
```

### 5.1.5 EtcBrwGetTitleString

a) Explanation of Commands: The title of current stage will be saved as variable.

b) Input Value : String Type Variable Name

c) Return Value : 1

d) Example

```
var titleName:String;
var value:Number = ext_fscommand2("EtcBrwGetTitleString", "titleName");
```

e) Example Explanation

If the current browser is set to the uppermost stage, '1' is saved to the 'value' variable and the 'list' string is saved to the 'titleName' variable.

### 5.1.6 EtcBrwGetCurIndex

a) Explanation of Commands : Returns the active index in the current browser list. It is recommended to call the FS Command below and then call this FS Command, subsequently processing the GUI for the currently active index.

- When reset the browser to 'EtcBrwSetInitialize'
- When browser stage is moved to lower level by 'EtcBrwSetNextStage'
- When browser stage is moved to higher level by 'EtcBrwSetPrevStage'

b) Input Value : None

c) Return Value : '0 ~'

d) Example

```
var value:Number = ext_fscommand2("EtcBrwGetCurIndex");
```

### 5.1.7 EtcBrwGetFolderNum

a) Explanation of Commands : Numbers of folders and tags from the current browser list will be returned.

b) Input Value : None

c) Return Value : '0 ~'

d) Example

```
var value:Number = ext_fscommand2("EtcBrwGetFolderNum");
```

### 5.1.8 EtcBrwAddFavorite

a) Explanation of Commands : Adds applicable files to the Music or Video 'favorites' according to the file types of the index input. This FS Command only works when the applicable index is 'folder', 'music file', or 'video file'.

b) Input Value : '0' ~ 'Total number of current browser - 1'

- '0' ~ ' Total number of current browser - 1' : Index that will be saved in 'Favorite'

c) Return Value : '-1 ~ 255'

- '-1' : If the maximum number of favorites is exceeded or the files could not be added properly to 'favorites' because, for example, the file type is not appropriate for 'favorites'.
- '0 ~ 255' : Returns the 'favorites index' value when the file is properly added to 'favorites'.

d) Example

```
var value:Number = ext_fscommand2("EtcBrwAddFavorite", "2");
```

### 5.1.9 EtcBrwDelete

a) Explanation of Commands: Deletes input index 'file' or 'folder'.

b) Input Value : '0' ~ 'Total number of current browser - 1'

- '0' ~ "Total number of current browser - 1" : Index to be deleted.

c) Return Value : '-1 ~'

- '-1' : If the applicable folder is the folder in which the recorded files are stored ('Recordings', 'FM Radio', 'Line-in', Mobile TV', 'Video Capture', 'Mic'), or the files are not properly deleted after attempting to delete them from the tag browser.
- '0 ~' : Returns the total number of the remaining browser list after deleting it properly.

d) Example

```
var value:Number = ext_fscommand2("EtcBrwDelete", "2");
```

### 5.1.10 EtcBrwGetType

a) Action Explanation : Return the current browser type. This function only works with 'Default', 'Picture' Browser.

b) Input Value : None

c) Output Value : '-1 ~ 1'

- '-1' : When it works with 'Favorite', 'Bookmark' Browser.
- '0' : When Browser is located in 'Folder'
- '1' : When Browser is located other than in 'Folder'

d) Example

```
var value:Number = ext_fscommand2("EtcBrwGetType");
```

## 5.2 Mode FS Command

Explains Mode related FS Command.

### 5.2.1 EtcModChangeMode

a) Explanation of Commands : Changes the Mode. Since the device does not operate properly if this command is not called between modes, this FS Command must be called when the Flash file that pertains to each Mode is first executed in order for the Low Level class to operate properly to each Mode. Only, because 'Flash' and 'Text' executes a single content without a playlist, this FS Command must be called in the browser. Please refer to the UCI structure documentation.

b) Input Value : String

- 'Music' : Switch Low Level Class to Music Mode
- 'Video' : Switch Low Level Class to Motion Picture Mode
- 'Radio' : Switch Low Level Class to FM Radio Mode
- 'Record' : Switch Low Level Class to Recording Mode
- 'MobileTV' : Switch Low Level Class to DMB/DAB Mode
- 'Flash' : Switch Low Level Class to Flash Mode
- 'Text' : Switch Low Level Class to Text Mode

- 'Picture' : Switch Low Level Class to Picture Mode
- 'Dictionary' : Switch Low Level Class to Dictionary (Domestic Only)

DO NOT APPLY in UCI

c) Return Value : '-1 ~ 1'

- '-1' : When input string is incorrect.
- '0' : When the currently operating mode is reset.
- '1' : When mode is converted in a normal way.

d) Example

```
var mode:String = "Music";
var value:Number = ext_fscommand2("EtcModChangeMode", mode);
```

e) Example Explanation

If the example above is executed in the Radio Mode, '1' is saved to the 'value' variable, and the Low Level class is changed to the Music Mode. If the example above is executed in the Music Mode, '0' is saved to the 'value' variable, and there is no separate processing in the Low Level class.

## 5.2.2 EtcModSetWidgetMode

a) Action Explanation : Setting Widget Mode. This FS Command's role is to handle the Widget function with the firmware. Therefore, one shall need to set 'On' when starts Widget with FS Command, and also need to set 'Off' when Widget is finished.

b) Input Value : '0 ~ 1'

- '0' : On
- '1' : Off

c) Output Value : '-1 ~ 1'

- '-1' : When the input value exceeds the limit or it can't work normally.
- '0 ~ 1' : Return the input value when it works normally

d) Example

```
var value:Number = ext_fscommand2("EtcModSetWidgetMode ", "0");
```

## 5.3 Document FS Command

Explaining FS Command related to Documents

### 5.3.1 EtcTxtOpen

a) Explanation of Commands : Opens a document file. The eligible documents are those that are selected from the browser. The 'Close' action takes place if 'EtcModChangeMode' FS Command is called or 'EtcTxtOpen' FS Command is called again. Document-related FS Commands guarantees normal operation after the document is opened.

b) Input Value : None

c) Return Value : '0 ~ 1'

- '0' : When the file is not opened normally.
- '1' : When the file is opened normally.

d) Example

```
var value:Number = ext_fscommand2("EtcTxtOpen");
```

### 5.3.2 EtcTxtRead

a) Explanation of Commands : Reads a portion of the document and saves it to the variable.

This FS Command must be called in the following cases:

- When the text is printed on the screen for the first time.
- When the current text contents is reloaded.

- When 'EtcTxtGotoOffset' FS Command is called.
- b) Input Value : String Type Variable Name
- c) Return Value : '0 ~ 1'
  - '0' : When normal operation did not occur due to no text contents.
  - '1' : When it properly reads the contents of documents
- d) Example
 

```
var data:String;
var value:Number = ext_fscommand2("EtcTxtRead", "data");
```

### 5.3.3 EtcTxtNextPage

- a) Explanation of Commands : Reads from the next page of current location and save as variable.
- b) Input Value : String Type Variable Name
- c) Return Value : '0 ~ 1'
  - '0' : When normal operation did not occur due to the current page being the last page.
  - '1' : When it properly reads the contents of documents
- d) Example
 

```
var data:String;
var value:Number = ext_fscommand2("EtcTxtNextPage", "data");
```

### 5.3.4 EtcTxtNextLine

- a) Explanation of Commands : Reads from the next line of current location and save as variable.
- b) Input Value : String Type Variable Name
- c) Return Value : '0 ~ 1'
  - '0' : When normal operation did not occur due to the current page being the last page.
  - '1' : When it properly reads the contents of documents
- d) Example
 

```
var data:String;
var value:Number = ext_fscommand2("EtcTxtNextLine", "data");
```

### 5.3.5 EtcTxtPrevPage

- a) Explanation of Commands : Read from the previous page of current location and save as variable.
- b) Input Value : String Type Variable Name
- c) Return Value : '0 ~ 1'
  - '0' : When the normal operation did not occur due to current page being the first page.
  - '1' : When it properly reads the contents of documents
- d) Example
 

```
var data:String;
var value:Number = ext_fscommand2("EtcTxtPrevPage", "data");
```

### 5.3.6 EtcTxtPrevLine

- a) Explanation of Commands : Reads from the previous line of current location and save as variable.
- b) Input Value : String Type Variable Name
- c) Return Value : '0 ~ 1'
  - '0' : When the normal operation did not occur due to current page being the first page.
  - '1' : When it properly reads the contents of documents
- d) Example
 

```
var data:String;
var value:Number = ext_fscommand2("EtcTxtPrevLine", "data");
```

### 5.3.7 EtcTxtGetTotalSize

- a) Explanation of Commands : Return Total Size

- b) Input Value : None
- c) Return Value : '0 ~'  
- '0 ~' : (unit) Byte
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtGetTotalSize");
```

### 5.3.8 EtcTxtGetCurOffset

- a) Explanation of Commands : Returns the beginning page point of the current text.
- b) Input Value : None
- c) Return Value : '0 ~'  
- '0 ~' : (unit) Byte
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtGetCurOffset");
```

### 5.3.9 EtcTxtGotoOffset

- a) Explanation of Commands : Moves the page starting point of the current document to the desired location. If the input value falls out of range, this command takes the first or the last page, calculates in the Low Level class, and then returns the value.
- b) Input Value : Offset value to be moved. (unit) Byte
- c) Return Value : '0 ~'  
- '0 ~' : (unit) Byte
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtGotoOffset", "1024");
```

### 5.3.10 EtcTxtGetOnePageSize

- a) Explanation of Commands : Returns the size of the first single page. If the document is short, there are times when the output fits into one page due to the small size of the total length ('EtcTxtGetTotalSize'). In such cases, exception handling is needed for functions such as scroll bars and shortcuts. Since the first single page is related to the font size, the size of the single page must be updated every time the font size is changed.
- b) Input Value : None
- c) Return Value : '0 ~'  
- '0 ~' : (unit) Byte
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtGetOnePageSize");
```

### 5.3.11 EtcTxtSetBookmark

- a) Explanation of Commands : Adds the current beginning page location of the current text to the text bookmark list.
- b) Input Value : None
- c) Return Value : '-1 ~ 255'  
- '-1' : When adding cannot occur due to text bookmark list number being full.  
- '0 ~ 255' : Returns the index value when added normally.
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtSetBookmark");
```

### 5.3.12 EtcTxtSetFontsize

- a) Explanation of Commands : Sets the font size on the text screen.

- b) Input Value : '1 ~'
  - '1 ~' : Font Size, (unit) Pixel
- c) Return Value : '1'
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtSetFontSize", "16");
```

### 5.3.13 EtcTxtSetLineCount

- a) Explanation of Commands : Sets the number of lines on the text screen.
- b) Input Value : '1 ~'
  - '1 ~' : The number of lines
- c) Return Value : '1'
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtSetLineCount", "24");
```

### 5.3.14 EtcTxtSetDisplayWidth

- a) Explanation of Commands : Set width of the window of documents
- b) Input Value : '1 ~'
  - '1 ~' : Width, (unit) Pixel
- c) Return Value : '1'
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtSetDisplayWidth", "246");
```

### 5.3.15 EtcTxtSetRepeatCount

- a) Explanation of Commands : Sets the number of lines being moved when moving to 'EtcTxtNextLine', 'EtcTxtPrevLine' FS Command. The value will be reset to '1' when 'EtcTxtNextLine', 'EtcTxtPrevLine' FS Commands are called.
- b) Input Value : '1 ~'
  - '1 ~' : The number of lines to be shifted
- c) Return Value : '1'
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtSetRepeatCount", "5");
```

### 5.3.16 EtcTxtGetLastPageOffset

- a) Explanation of Commands : Returns the starting location of the last page. Since the starting location of the last page is related to the font size, the starting location value of the last page must be updated every time the font size is changed.
- b) Input Value : None
- c) Return Value : '0 ~'
  - '0 ~' : (unit) Byte
- d) Example

```
var value:Number = ext_fscommand2("EtcTxtGetLastPageOffset");
```

## 5.4 User Data FS Command

Explanation of User Data FS Commands : User Data FS Command provides 10 constant number storage spaces and 10 character storage spaces (up to 256 bytes), as well as one constant number storage space related to the main menu settings. If the 'PARAM.CFG' file in the 'System' folder as in formatting the device, or the settings values become initialized through the 'Restore System Default' function ('SetSysDefault'), the constant number storage space is initialized

to constant '0' and the character storage spaces are initialized to "Null".

#### 5.4.1 EtcUsrGetNumber

- a) Explanation of Commands : Obtains the constant variable value of the applicable index.
- b) Input Value : '0 ~ 9'
  - '0 ~ 9' : Index
- c) Return Value : '~'
  - '-1' : If the index input falls out of range or the saved value is '-1'
  - '~' : Returns a positive number saved in the applicable index after normal processing
- d) Example

```
var value:Number = ext_fscommand2("EtcUsrGetNumber", "4");
```
- e) Example Explanation

Returns a constant variable value saved in index '4' to the 'value' variable.

#### 5.4.2 EtcUsrSetNumber

- a) Explanation of Commands : Save the punched positive number in the pertinent index
- b) Input Value : '0 ~ 9', '~'
  - First Input Value
    - '0 ~ 9' : Index
  - Second Input Value
    - '~' : A positive number to be saved
- c) Return Value : '-1', '1'
  - '-1' : When the punch index exceeds the limit
  - '1' : When it saves a positive number in the index
- d) Example

```
var value:Number = ext_fscommand2("EtcUsrSetNumber", "2", "1000");
```
- e) Example Explanation

Return '1' to 'value' variable and save '1000' positive number at 2 index

#### 5.4.3 EtcUsrGetString

- a) Explanation of Commands : Save the pertinent index's string in the inputted variable
- b) Input Value : '0 ~ 9', String Type Variable Name
  - First Input Value
    - '0 ~ 9' : Index
  - Second Input Value
    - The name of variables to be saved as an array of strings
- c) Return Value : '-1', '1'
  - '-1' : When the punched index exceeds the limit
  - '1' : When it works normally
- d) Example

```
var userData:String;
var value:Number = ext_fscommand2("EtcUsrGetString", "4", "userData");
```
- e) Example Explanation

Save '1' in 'value' variable. And the string of 4 index is saved in 'userData' variable

#### 5.4.4 EtcUsrSetString

- a) Explanation of Commands : Save String at the pertinent index

b) Input Value : '0 ~ 9', An array of strings

First Input Value

- '0 ~ 9' : index

Second Input Value

- The array of strings to be saved

c) Return Value : '-1', '1'

- '-1' : When it cannot be processed normally such as when the index input is out of the range or the length of the string is greater than 256 bytes.

- '1' : When it works normally

d) Example

```
var userData:String = "ABCDEFGH";
```

```
var value:Number = ext_fscommand2("EtcUsrSetString", "2", userData);
```

e) Example Explanation

Save '1' in 'value' variable. And 'ABCDEFGH' string is saved in '2' index

#### 5.4.5 EtcUsrGetMainmenu

a) Explanation of Commands : Returns the main menu category value that is currently set. This FS Command is used in the launcher class and for the Load\_SWF( ) function.

b) Input Value : None

c) Return Value : The value of a positive number

d) Example

```
var value:Number = ext_fscommand2("EtcUsrGetMainmenu");
```

#### 5.4.6 EtcUsrSetMainmenu

a) Explanation of Commands: Set Main Menu Type. The loaded Mainmenu according to values is as follows.

- 0 : Load 'mainmenu1.swf'

- 1 : Load 'mainmenu2.swf'

- 2 : Load mainmenu3.swf'

- 3 : Load 'widget.swf'

- Other Values : Load 'mainmenu1.swf'

b) Input Value : A positive number

- A positive number: The value of a main menu type

c) Return Value : A positive number

- A positive number : Return Input Value

d) Example

```
var value:Number = ext_fscommand2("EtcUsrSetMainmenu", "0");
```

e) Example Explanation

'0' is saved into the 'value' variable and 'mainmenu1.swf' is loaded upon calling the Load-SWF(\_global.MODE\_MAIN), Load-SWF(\_global.MODE\_MAIN2), and Load-SWF(\_global.MODE\_MAIN3) functions.

### 5.5 Notepad FS Command

It explains FS Command about Notepad.

#### 5.5.1 EtcNpdSave

a) Explanation of Commands: save specific area in current page as notepad file in input index. In COWON S9, save square area that formed as [0, 34], [271, 34], [0, 431], [271, 431]. It is possible to load a saved page by loadMoive name of '0.NPM' ~ '39.NPM'. if this FS Command is called then notepad buffer that created by FS Command

EtcNpdCaptureBuffer will be deleted automatically.

b) Input Value: '0 ~39'

- '0 ~ 39': notepad index that will be saved.

c) Return Value: '-1 ~ 39'

- '-1': If it doesn't work properly because the input value has exceeded the limit.

d) Example

```
var value:Number = ext_fscommand2("EtcNpdSave", "3");
```

e) Explanation of example

Save value '3' as variable value and save specific area in current page as a notepad file that contains index 3. Only thing is that this FS command is called while frame is working, it has to be called in the next frame after save current page that you want to save.

### 5.5.2 EtcNpdCheck

a) Explanation of Commands: check if there are any notepad files exist in the input index.

b) Input Value: '0 ~ 39'

- '0 ~ 39': Notepad index that you want to check if it exists.

c) Return Value: '-1 ~39'

- '-1': if the file doesn't exist or the input value has exceeded the limit.

- '0 ~ 39': return the value of index if there is a file in the index.

d) Example

```
var value:Number = ext_fscommand2("EtcNpdCheck", "3");
```

e) Explanation of example

If there is a notepad file in third index then save '3' as variable value.

### 5.5.3 EtcNpdDelete

a) Explanation of Commands: delete notepad files that are saved.

b) Input Value: '0 ~ 39'

- '0 ~ 39' : index of notepad that will be deleted.

c) Return Value: '-3 ~39'

- '-1' : If it doesn't work properly because the input value has exceeded the limit.

- '0 ~39' : return the input index value if it worked properly.

d) Example

```
var value:Number = ext_fscommand2("EtcNpdDelete", "5");
```

e) Explanation of example

Save 5 as variable value and delete the file, if there is a notepad file that exists in fifth line in the index.

### 5.5.4 EtcNpdCaptureBuffer

a) Explanation of Commands: create temporary buffer in the index and save current page in the temporary buffer. If there is a temporary buffer already existed then save current page in the temporary buffer only. In COWON S9, save square area that formed as [0, 34], [271, 34], [0, 431], [271, 431]. It is possible to load a saved page by loadMovie name of '0.NPM' ~ '1.NPM'. Because it is using internal memory, if you don't need temporary buffer then delete buffer by 'EtcNpdClearBuffer' FS Command before you turn off the device. .

b) Input Value: '0 ~1'

- '0 ~ 1': index of temporary buffer that will be saved.

c) Return Value: '-1 ~ 1'

- '-1': If it doesn't work properly because the input value has exceeded the limit.

- '0 ~ 1': return the value of index if temporary buffer is saved properly.

d) Example

```
var value:Number = ext_fscommand2("EtcNpdCaptureBuffer", "0");
```

e) Explanation of example

Save '0' as variable value and save current page as index 0 in temporary buffer. However, because FS Command is called while frame runs, you have to call next frame after current frame is saved.

### 5.5.5 EtcNpdCheckBuffer

a) Explanation of Commands: check if there is temporary buffer exists in the input index.

b) Input Value: '0 ~ 1'

- '0 ~ 1': note pad index that you want to know if it exists.

c) Return Value: '-1 ~ 1'

- '-1': if input value is exceed of the limit or temporary buffer doesn't exit.

- '0 ~ 1': return the input value if temporary buffer exists.

d) Example

```
var value:Number = ext_fscommand2("EtcNpdCheckBuffer", "1");
```

e) Explanation of example

Save '1' as variable value if temporary buffer exists in the first index.

### 5.5.6 EtcNpdClearBuffer

a) Explanation of Commands: delete the temporary buffer in the index.

b) Input Value: '0 ~ 2'

- '0 ~ 1': index of temporary buffer that will be deleted.

- '2': if you want to delete all temporary buffers that were created.

c) Return Value : '-1 ~ 2'

- '-1': if it doesn't work properly because of the input value is exceed the limit or there is no temporary buffer in the index.

- '0 ~ 2': Return the input value if it worked properly.

d) Example

```
var value:Number = ext_fscommand2("EtcNpdClearBuffer", "2");
```

e) Explanation of example

Save variable value as '2' and delete all temporary buffers that were created.