

Package	flash.filesystem
Class	File
Public Properties	applicationDirectory : File applicationStorageDirectory : File desktopDirectory : File documentsDirectory : File exists : Boolean icon : Icon isDirectory : Boolean isHidden : Boolean isPackage : Boolean isSymbolicLink : Boolean lineEnding : String nativePath : String parent : File separator : String spaceAvailable : Number systemCharset : String url : String userDirectory : File
Public Methods	File(path:String = null) browseForDirectory(title:String):void browseForOpen(title:String, typeFilter:Array = null):void browseForOpenMultiple(title:String, typeFilter:Array = null):void browseForSave(title:String):void cancel():void canonicalize():void clone():File copyTo(newLocation:FileReference, overwrite:Boolean = false):void copyToAsync(newLocation:FileReference, overwrite:Boolean = false):void createDirectory():void createTempDirectory():File createTempFile():File deleteDirectory(deleteDirectoryContents:Boolean = false):void deleteDirectoryAsync(deleteDirectoryContents:Boolean = false):void deleteFile():void deleteFileAsync():void getDirectoryListing():Array getDirectoryListingAsync():void getRelativePath(ref:FileReference, useDotDot:Boolean = false):String getRootDirectories():Array moveTo(newLocation:FileReference, overwrite:Boolean = false):void moveToAsync(newLocation:FileReference, overwrite:Boolean = false):void moveToTrash():void moveToTrashAsync():void resolvePath(path:String):File
Events	cancel complete directoryListing ioError securityError select selectMultiple

Package	flash.filesystem
Class	FileStream
Public Properties	bytesAvailable : uint endian : String objectEncoding : uint
Public Methods	readAhead : Number FileStream() close():void open(file:File, fileMode:String):void openAsync(file:File, fileMode:String):void readBoolean():Boolean readByte():int readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void readDouble():Number readFloat():Number readInt():int readMultiByte(length:uint, charSet:String):String readObject():* readShort():int readUnsignedByte():uint readUnsignedInt():uint readUnsignedShort():uint readUTF():String readUTFBytes(length:uint):String
Events	readUTFBytes(length:uint):String close complete ioError outputProgress progress

Package	flash.security
Class	XMLSignatureValidator
Public Properties	digestStatus : String identityStatus : String referencesStatus : String referencesValidationSetting : String revocationCheckSetting : String signerCN : String signerDN : String signerExtendedKeyUsages : Array signerTrustSettings : Array uriDereferencer : IURIDereferencer useSystemTrustStore : Boolean validityStatus : String
Public Methods	XMLSignatureValidator() addCertificate(cert:ByteArray, trusted:Boolean):* verify(signature:XML):void
Events	complete error

Package	flash.net.drm
Class	DRMVoucher
Public Properties	offlineLeaseEndDate : Date offlineLeaseStartDate : Date playbackTimeWindow : DRMPlaybackTimeWindow policies : Object voucherEndDate : Date voucherStartDate : Date

Package	flash.net.drm
Class	DRMContentData
Public Properties	authenticationMethod : String domain : String licenseID : String serverURL : String

Package	flash.net
Class	URLRequestDefaults
Public Properties	authenticate : Boolean cacheResponse : Boolean followRedirects : Boolean manageCookies : Boolean useCache : Boolean userAgent : String

Package	flash.net.drm
Class	DRMPlaybackTimeWindow
Public Properties	endDate : Date period : uint startDate : Date

Package	Package: flash.errors
Class	DRMManagerError
Public Properties	subErrorID : int
Public Methods	DRMManagerError(message:String, id:int, subErrorID:int) toString():String

Package	flash.net.drm
Class	DRMManager
Public Methods	authenticate(serverURL:String, domain:String, username:String, password:String):void getDRMManager():DRMManager loadVoucher(contentData:DRMContentData) resetDRMVouchers():void setAuthenticationToken(serverURL:String, domain:String, token:ByteArray):void

Package	flash.errors
Class	SQLError
Public Properties	detailArguments : Array detailID : int details : String operation : String
Public Methods	SQLError(operation:String, details:String = "", message:String = "", id:int = 0, detailID:int = -1, detailArgs:Array = null) toString():String