

Class	SQLConnection
Public Properties	autoCompact : Boolean cacheSize : uint columnNameStyle : String connected : Boolean inTransaction : Boolean lastInsertRowID : Number pageSize : uint totalChanges : Number
Public Methods	SQLConnection() addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void analyze(resourceName:String = null, responder:Responder = null):void attach(name:String, reference:Object = null, responder:Responder = null, encryptionKey:ByteArray = null):void begin(option:String = null, responder:Responder = null):void cancel(responder:Responder = null):void close(responder:Responder = null):void commit(responder:Responder = null):void compact(responder:Responder = null):void deanalyze(responder:Responder = null):void detach(name:String, responder:Responder = null):void getSchemaResult():ISchemaResult loadSchema(type:Class = null, name:String = null, database:String = "main", includeColumnSchema:Boolean = true, responder:Responder = null):void open(reference:Object = null, openMode:String = "create", autoCompact:Boolean = false, pageSize:int = 1024, encryptionKey:ByteArray = null):void openAsync(reference:Object = null, openMode:String = "create", responder:Responder = null, autoCompact:Boolean = false, pageSize:int = 1024, encryptionKey:ByteArray = null):void reencrypt(newEncryptionKey:ByteArray, responder:Responder = null):void removeEventListener(type:String, listener:Function, useCapture:Boolean = false):void
Events	rollback analyze attach begin cancel close commit compact deanalyze delete detach error insert open reencrypt rollback schema update

Class	SQLSchema
Public Properties	database : String name : String sql : String
Public Methods	SQLSchema(database:String, name:String, sql:String)

Class	SQLTableSchema
Public Properties	columns : Array
Public Methods	SQLTableSchema(database:String, name:String, sql:String, columns:Array)

Class	EncryptedLocalStore
Public Methods	getItem(name:String):ByteArray removeItem(name:String):void reset():void setItem(name:String, data:ByteArray, stronglyBound:Boolean = false):void

Class	SQLColumnSchema
Public Properties	allowNull : Boolean autoIncrement : Boolean dataType : String defaultCollationType : String name : String primaryKey : Boolean
Public Methods	SQLColumnSchema(name:String, primaryKey:Boolean, allowNull:Boolean, autoIncrement:Boolean, dataType:String, defaultCollationType:String)

Class	SQLIndexSchema
Public Properties	table : String
Public Methods	SQLIndexSchema(database:String, name:String, sql:String, table:String)

Class	SQLViewSchema
Public Methods	SQLViewSchema(database:String, name:String, sql:String, columns:Array)

Class	SQLSchemaResult
Public Properties	indices : Array tables : Array triggers : Array views : Array
Public Methods	SQLSchemaResult(tables:Array, views:Array, indices:Array, triggers:Array)

Class	SQLResult
Public Properties	complete : Boolean data : Array lastInsertRowID : Number rowsAffected : Number
Public Methods	SQLResult(data:Array = null, rowsAffected:Number = 0, complete:Boolean = true, rowID:Number = 0)

Class	SQLTriggerSchema
Public Properties	table : String
Public Methods	SQLTriggerSchema(database:String, name:String, sql:String, table:String)

Class	SQLStatement
Public Properties	executing : Boolean itemClass : Class parameters : Object sqlConnection : SQLConnection text : String
Public Methods	SQLStatement() cancel():void clearParameters():void execute(prefetch:int = -1, responder:Responder = null):void getResult():SQLResult next(prefetch:int = -1, responder:Responder = null):void
Events	error result