



LIBRERIA Tfhkaif

MANUAL DE INTEGRACION

Versión 1.1
Mayo de 2017

The Factory HKA, C.A.

MANUAL DE INTEGRACION - LIBRERÍA Tfhkaif

VERSIÓN 1.1

The Factory HKA
Departamento de Soporte e Integración
integration@thefactoryhka.com

Historial de revisiones

Revisión	Fecha	Paginas Afectadas	Comentario
1.0	02/05/2014	Todas	Versión inicial
1.1	19/05/2017	Todas	Se actualiza información general

Contenido

Historial de revisiones	7
Introducción.....	5
Requerimientos de hardware.....	5
Requerimientos de software	5
Importación y declaraciones	6
Funciones de la Librería Tfhkaif.....	9
OpenFpctrl.....	9
CloseFpctrl.....	9
CheckFprinter	9
ReadFpStatus.....	10
SendCmd	10
SendFileCmd.....	10
UploadReportCmd.....	11
UploadStatusCmd.....	11
UploadReportDin.....	12
UploadStatusDin.....	12
Anexos.....	13
Anexo 1: Lista de códigos de Status	13
Anexo 2: Lista de códigos de Error	14

Introducción

La librería “Tfhkaif.dll” permite la integración con sistemas administrativos desarrollados bajo la tecnología de Win 32; como lo son: Delphi, FoxPro, PowerBuilder y Visual Basic 6.

En las siguientes páginas se describen los detalles de la interfaz de aplicación “Tfhkaif.dll”, desde los requerimientos de hardware, la referencia a la librería de integración, los componentes adicionales tales como librerías dinámicas y finalmente la estructura de la Clase compilada que contiene las funciones para enviar comandos e interactuar con cualquiera de las impresoras distribuidas por The Factory HKA a través de un sistema administrativo desarrollado bajo Win 32.

Requerimientos de hardware

- Procesador de 1Ghz o superior
- Memoria RAM de 512MB o superior
- Espacio libre en Disco Duro de 50MB
- Puerto Serial físico o USB 2.0

Requerimientos de software

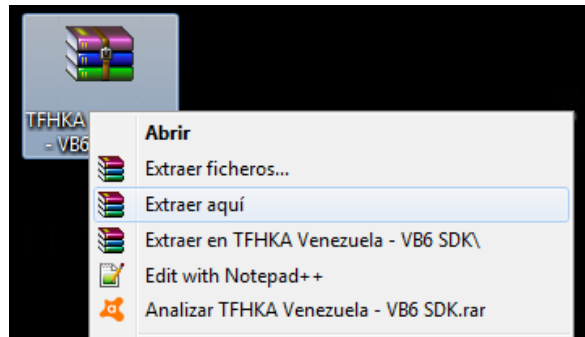
- Sistema Operativo Windows
- IDE para manejar un desarrollo en Delphi, FoxPro, PowerBuilder y Visual Basic 6

Importación y declaraciones

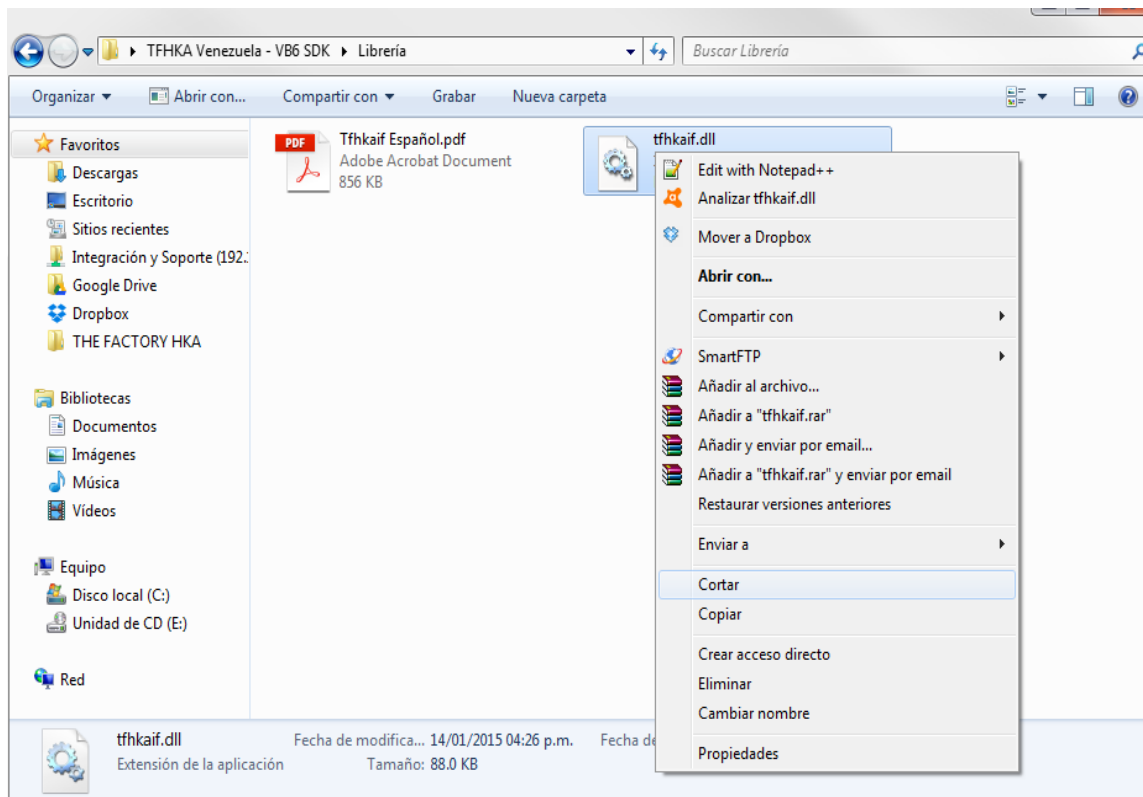
En el SDK descargado, usted se encontrará con las tres herramientas básicas para realizar la integración del sistema administrativo con nuestra impresora fiscal; la librería de integración “Tfhkaif.dll”, el manual de dicha librería y un demo funcional con código abierto donde se ejemplifica el uso de la librería de forma práctica.

Antes de comenzar a hacer uso de la librería en el desarrollo, lo primero que se debe hacer es incluir y referenciar la librería en nuestro proyecto. Esto se consigue siguiendo los siguientes pasos:

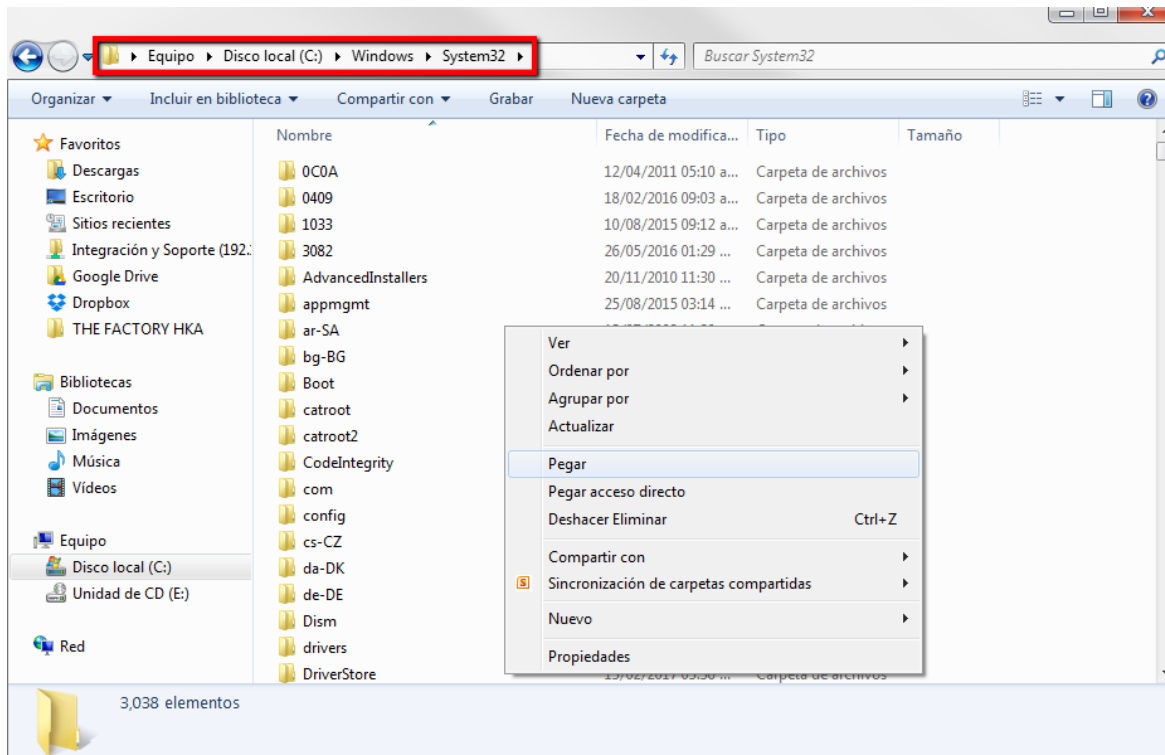
- Descomprimir el contenido del SDK descargado.



- Acceder a la carpeta SDK descomprimida previamente y copiar la librería “Tfhkaif.dll” que se encuentra dentro de la carpeta Demo o en la carpeta Library.



- Pegar la librería en la carpeta System32 de Windows y en la carpeta donde se encuentra del ejecutable del proyecto.



- Finalmente, crear las variables de control para trabajar la librería durante el uso de todo el desarrollo y declarar las distintas funciones que componen la librería.

➤ En Visual Basic 6:

'Funciones para manejar la Impresora Fiscal

```
Declare Function OpenFpctrl Lib "tfhkaif.dll" (ByVal lpPortName As String) As Boolean
Declare Function CloseFpctrl Lib "tfhkaif.dll" () As Boolean
Declare Function CheckFprinter Lib "tfhkaif.dll" () As Boolean
Declare Function ReadFpStatus Lib "tfhkaif.dll" (Status As Long, Error As Long) As Boolean
Declare Function SendCmd Lib "tfhkaif.dll" (Status As Long, Error As Long, ByVal cmd As String) As Boolean
Declare Function SendFileCmd Lib "tfhkaif.dll" (Status As Long, Error As Long, ByVal file As String) As Long
Declare Function UploadReportCmd Lib "tfhkaif.dll" (Status As Long, Error As Long, ByVal cmd As String, ByVal file As String) As Boolean
Declare Function UploadStatusCmd Lib "tfhkaif.dll" (Status As Long, Error As Long, ByVal cmd As String, ByVal file As String) As Boolean
Declare Function UploadReportDin Lib "tfhkaif.dll" (Status As Long, Error As Long, ByVal cmd As String, ByVal Cadena As String) As Boolean
Declare Function UploadStatusDin Lib "tfhkaif.dll" (Status As Long, Error As Long, ByVal cmd As String, ByVal Cadena As String) As Boolean
```

'Variables de Control

```
Public Respuesta As Boolean
Public Status As Long
Public Error As Long
```

➤ En Visual FoxPro:

*Declaración de la DLL

```
DECLARE INTEGER OpenFpctrl IN TFHKAIF.DLL String lpPortName
DECLARE INTEGER CloseFpctrl IN TFHKAIF.DLL
DECLARE INTEGER CheckFprinter IN TFHKAIF.DLL
DECLARE INTEGER ReadFpStatus IN TFHKAIF.DLL LONG @lStatus ,LONG @lError
DECLARE INTEGER SendCmd IN TFHKAIF.DLL LONG @lStatus ,LONG @lError,STRING @cmd
DECLARE INTEGER SendFileCmd IN TFHKAIF.DLL LONG @lStatus ,LONG @lError,STRING @fileCmd
DECLARE INTEGER UploadReportCmd IN TFHKAIF.DLL LONG @lStatus ,LONG @lError,STRING @cmd,string @fileCmd
DECLARE INTEGER UploadStatusCmd IN TFHKAIF.DLL LONG @lStatus ,LONG @lError,STRING @cmd,string @fileCmd
DECLARE INTEGER UploadReportDin IN TFHKAIF.DLL LONG @lStatus ,LONG @lError,STRING @cmd,STRING @cadenaCmd
DECLARE INTEGER UploadStatusDin IN TFHKAIF.DLL LONG @lStatus ,LONG @lError,STRING @cmd,STRING @cadenaCmd
```

*Declaración variables de control

```
PUBLIC lStatus, lError
PUBLIC Respuesta
```

➤ En Delphi:

var

```
Respuesta : Boolean;
Stat, Err : Integer;
```

```
function OpenFpctrl(sPort:PChar): Boolean;stdcall;
function CloseFpctrl(): Boolean;stdcall;
function CheckFprinter(): Boolean;stdcall;
function ReadFpStatus(var status, error : Integer): Boolean;stdcall;
function SendCmd(var status, error : Integer; cmd : PChar): Integer;stdcall;
function SendFileCmd(var status, error : Integer; sfile : PChar): Integer;stdcall;
function UploadReportCmd(var status, error : Integer; rTipo, sFile : PChar): Boolean;stdcall;
function UploadStatusCmd(var status, error : Integer; sTipo, sFile : PChar): Boolean;stdcall;
function UploadReportDin(var status, error : Integer; rTipo : PChar; Cadena : PChar): Boolean;stdcall;
function UploadStatusDin(var status, error : Integer; sTipo : PChar; Cadena : PChar): Boolean;stdcall;
```

implementation

```
function OpenFpctrl;external 'TfhkaIf.dll' name 'OpenFpctrl';
function CloseFpctrl;external 'TfhkaIf.dll' name 'CloseFpctrl';
function CheckFprinter;external 'TfhkaIf.dll' name 'CheckFprinter';
function ReadFpStatus;external 'TfhkaIf.dll' name 'ReadFpStatus';
function SendCmd;external 'TfhkaIf.dll' name 'SendCmd';
function SendFileCmd;external 'TfhkaIf.dll' name 'SendFileCmd';
function UploadReportCmd;external 'TfhkaIf.dll' name 'UploadReportCmd';
function UploadStatusCmd;external 'TfhkaIf.dll' name 'UploadStatusCmd';
function UploadReportDin;external 'TfhkaIf.dll' name 'UploadReportDin';
function UploadStatusDin;external 'TfhkaIf.dll' name 'UploadStatusDin';
```


➤ **En Power Builder:**

```
//Variables de Control
Long ll_respuesta
Long ll_status
Long ll_error1

//Funciones para la impresora Fiscal
Function Long OpenFpctrl( String IpPortName) LIBRARY "tfhkaif.dll" alias for "OpenFpctrl;Ansi"
Function Long CloseFpctrl( ) LIBRARY "tfhkaif.dll" alias for "CloseFpctrl;Ansi"
Function Long CheckFprinter( ) LIBRARY "tfhkaif.dll" alias for "CheckFprinter;Ansi"
Function Long ReadFpStatus ( Ref Long status, Ref Long error1) LIBRARY "tfhkaif.dll" alias for "ReadFpStatus;Ansi"
Function Long SendCmd ( Ref Long status, Ref Long error1, String cmd) LIBRARY "tfhkaif.dll" alias for "SendCmd;Ansi"
Function Long SendFileCmd ( Ref Long status, Ref Long error1, String file) LIBRARY "tfhkaif.dll" alias for "SendFileCmd;Ansi"
Function Long UploadReportCmd (Ref Long status, Ref Long error1, String cmd, String file) LIBRARY "tfhkaif.dll" alias for "UploadReportCmd;Ansi"
Function Long UploadStatustCmd (Ref Long status, Ref Long error1, String cmd, String file) LIBRARY "tfhkaif.dll" alias for "UploadStatusCmd;Ansi"
Function Long UploadReportDin (Ref Long status, Ref Long error1, String cmd, ref String cadena) LIBRARY "tfhkaif.dll" alias for "UploadReportDin;Ansi"
Function Long UploadStatusDin (Ref Long status, Ref Long error1, String cmd, ref String cadena) LIBRARY "tfhkaif.dll" alias for "UploadStatusDin;Ansi"
```

Acto seguido, nos encontramos listos para empezar a utilizar las funciones que contiene la librería y poder enviar comandos a la impresora fiscal a través de la API.

Funciones de la Librería Tfhkaif

OpenFpctrl

Permite realizar la apertura del puerto de comunicaciones por el cual se establecerá comunicación con la impresora.

BOOLEAN OpenFpctrl(String IpPortName)

▪ **Parámetros:**

String IpPortName: Nombre del puerto COM a abrir.

▪ **Retorno:**

True: Puerto Abierto.

False: Falla en apertura del puerto.

CloseFpctrl

Permite cerrar el puerto COM abierto anteriormente:

VOID CloseFpctrl()

CheckFprinter

Permite verificar si la impresora está conectada al PC.

BOOLEAN CheckFprinter()

- **Retorno:**

True: Impresora conectada.

False: Impresora no detectada.

ReadFpStatus

Permite leer las variables de estado y error de la impresora. Al ejecutar este método se establece el valor de la variable tipo String *Estado*. (Ver Anexos 1 y 2)

BOOLEAN ReadFpStatus()

- **Retorno:**

True: Método ejecutado exitosamente

False: Error en ejecución del método

SendCmd

Permite realizar el envío de comandos hacia la impresora, en forma de tramas de caracteres ASCII, tal como es descrito en los manuales de integración de las respectivas impresoras y en el Manual de Protocolos y Comandos del protocolo TFHKA.

BOOLEAN SendCmd(Status Long, Error Long, String Cmd)

- **Parámetros:**

String Cmd: Trama en ASCII que se enviará a la impresora.

- **Retorno:**

True: Método ejecutado exitosamente.

False: Error en ejecución del método.

SendFileCmd

Permite realizar el envío de un archivo de lotes de comando hacia la impresora.

INT SendFileCmd(Status Long, Error Long, String rutaFile)

- **Parámetros:**

String rutaFile: ruta en la cual se encuentra el archivo de comandos por lote que se enviará a la impresora.

- **Retorno:**

INT, es el número de líneas del archivo de lotes de comandos ejecutadas exitosamente.

UploadReportCmd

Esta función permite almacenar en un archivo de texto simple un reporte de la impresora.

BOOLEAN UploadReportCmd(Status Long, Error Long, String cmd, String file)

▪ **Parámetros:**

String cmd: tipo de reporte a solicitar. Los reportes que pueden ser solicitados son los siguientes (consultar Manual de Protocolos y Comandos):

- En Venezuela los comandos para reportes son: U0X, U0Z, U1Z, U1X.
- En Panamá los comandos para reportes son: U0X, U0Z, U1Z, U1X.
- En República Dominicana los comandos para reportes son: U0Z01, U0Z02, U0Z03, U0Z04, U0X01, U0X02, U0X03, U0X04.
- En Curazao los comandos para reportes son: U0Z01, U0Z02, U0Z03, U0Z04, U0Z05, U0Z06, U0X01, U0X02, U0X03, U0X04, U0X05, U0X06.

String file: ruta del archivo donde se almacenará el reporte solicitado.

▪ **Retorno:**

True: Método ejecutado exitosamente.

False: Error en ejecución del método.

UploadStatusCmd

Esta función permite almacenar en un archivo de texto simple un status de la impresora.

BOOLEAN UploadStatusCmd(Status Long, Error Long, String cmd, String file)

▪ **Parámetros:**

String cmd: tipo de Status a solicitar. Los status que pueden ser solicitados son los siguientes (consultar Manual de Protocolos y Comandos):

- En Venezuela los comandos para status son: S1, S2, S3, S4, S5, S8E, S8P.
- En Panamá los comandos para status son: S1, S2, S3, S4, S5, S8E, S8P.
- En República Dominicana los comandos para status son: S1, S2, S3, S4, S5.
- En Curazao los comandos para status son: S1, S2, S3, S4, S5.

String file: ruta del archivo donde se almacenará el Status solicitado.

▪ **Retorno:**

True: Método ejecutado exitosamente.

False: Error en ejecución del método.

UploadReportDin

Esta función permite almacenar en una cadena un reporte de la impresora.

BOOLEAN UploadReportDin(Status Long, Error Long, String cmd, String cadena)

▪ **Parámetros:**

String cmd: tipo de reporte a solicitar. Los reportes que pueden ser solicitados son los siguientes (consultar Manual de Protocolos y Comandos):

- En Venezuela los comandos para reportes son: U0X, U0Z, U1Z, U1X.
- En Panamá los comandos para reportes son: U0X, U0Z, U1Z, U1X.
- En República Dominicana los comandos para reportes son: U0Z01, U0Z02, U0Z03, U0Z04, U0X01, U0X02, U0X03, U0X04.
- En Curazao los comandos para reportes son: U0Z01, U0Z02, U0Z03, U0Z04, U0Z05, U0Z06, U0X01, U0X02, U0X03, U0X04, U0X05, U0X06.

String cadena: variable donde se almacenará el reporte solicitado.

▪ **Retorno:**

True: Método ejecutado exitosamente.

False: Error en ejecución del método.

UploadStatusDin

Esta función permite almacenar en una cadena un status de la impresora.

BOOLEAN UploadStatusDin(Status Long, Error Long, String cmd, String cadena)

▪ **Parámetros:**

String cmd: tipo de Status a solicitar. Los status que pueden ser solicitados son los siguientes (consultar Manual de Protocolos y Comandos):

- En Venezuela los comandos para status son: S1, S2, S3, S4, S5, S8E, S8P.
- En Panamá los comandos para status son: S1, S2, S3, S4, S5, S8E, S8P.
- En República Dominicana los comandos para status son: S1, S2, S3, S4, S5.
- En Curazao los comandos para status son: S1, S2, S3, S4, S5.

String cadena: variable donde se almacenará el Status solicitado.

▪ **Retorno:**

True: Método ejecutado exitosamente.

False: Error en ejecución del método.

Anexo 1: Lista de códigos de Status

STATUS		
Retorno (Hex)	Retorno (Decimal)	Comentario
0	0	Estado desconocido.
1	1	En modo prueba y en espera.
2	2	En modo prueba y emisión de documentos fiscales.
3	3	En modo prueba y emisión de documentos no fiscales.
4	4	En modo fiscal y en espera.
5	5	En modo fiscal y emisión de documentos fiscales.
6	6	En modo fiscal y emisión de documentos no fiscales.
7	7	En modo fiscal, cercana carga completa de la memoria fiscal y en espera.
8	8	En modo fiscal, cercana carga completa de la memoria fiscal y en emisión de documentos fiscales.
9	9	En modo fiscal, cercana carga completa de la memoria fiscal y en emisión de documentos no fiscales.
0A	10	En modo fiscal, carga completa de la memoria fiscal y en espera.
0B	11	En modo fiscal, carga completa de la memoria fiscal y en emisión de documentos fiscales.
0C	12	En modo fiscal, carga completa de la memoria fiscal y en emisión de documentos no fiscales.

Nota: La librería “TfhkaNet.dll” retorna éste valor en decimal.

Anexo 2: Lista de códigos de Error

ERROR		
Retorno (Hex)	Retorno (Decimal)	Comentario
00	0	No hay error.
01	1	Fin en la entrega de papel.
02	2	Error de índole mecánico en la entrega de papel.
03	3	Fin en la entrega de papel y error mecánico.
50	80	Comando invalido o valor invalido.
54	84	Tasa invalida.
58	88	No hay asignadas directivas.
5C	92	Comando invalido.
60	96	Error fiscal.
64	100	Error de la memoria fiscal.
6C	108	Memoria fiscal llena.
70	112	Buffer completo. (debe enviar el comando de reinicio)
80	128	Error en la comunicación.
89	137	No hay respuesta.
90	144	Error LRC.
91	145	Error interno api.
99	153	Error en la apertura del archivo.

Nota: La librería "TfhkaNet.dll" retorna éste valor en decimal.