



# Functions Guide

Revised January 6, 2014

# Table of Contents

I. INTRODUCTION .....	3
II. FUNCTION DESCRIPTIONS .....	4
PCES_CaptureSignature( BoundData ) .....	4
PCES_GetOperatingMode .....	4
PCES_Register( ServerName ; ServerPort ; ServerPage ; licenseID ) .....	5
PCES_RenderSigFromString( SigString ; BoundData ) .....	5
PCES_SaveSigToFile( SigString ; BoundData ; Path ) .....	6
PCES_SetBannerPath( BMPLocation ) .....	6
PCES_SetImagePenWidth( Pixels ) .....	7
PCES_SetTabletComPort( PortNum ) .....	7
PCES_SetTabletMode( Mode ) .....	8
PCES_Version( type ) .....	8
III. RETURN/ERROR CODES.....	9
IV. CONTACT US .....	10

## **I. Introduction**

### **Description**

The eSign Signature Capture plug-in from Productive Computing offers functions that support capturing and rendering ESIGN / UETA compliant digital signatures. This plug-in allows FileMaker® Pro to capture signatures and bind them to information stored in your database. The current version of the plug-in is intended to be used with Topaz® 1x5 Signature Pads. With this plug-in a FileMaker user can capture and store legally binding signatures directly in a FileMaker solution. These operations are accomplished by using FileMaker function calls from within FileMaker calculations. These calculations are generally determined from within FileMaker 'SetField' or 'If' script steps.

### **Intended Audience:**

FileMaker developers or persons who have knowledge of FileMaker scripting, calculations and relationships as proper use of the plug-in requires that FileMaker integration scripts be created in your FileMaker solution.

### **Successful Integration Practices:**

- 1) Read the Developer's Guide
- 2) Read the Functions Guide
- 3) Review our FileMaker Demo

### **Technical Note**

Congress enacted the Electronic Signatures in Global and National Commerce Act (ESIGN Act) in June of 2000. The National Conference of Commissioners on Uniform State Laws (NCCUSL) adopted the Uniform Electronic Transactions Act (UETA) in 1999. Both Acts provide a legal framework for electronic transactions. Either Act gives electronic signatures and records the same validity and enforceability as manual signatures and paper-based transactions.

Before implementing electronic signatures and transactions in your organization consult an attorney to assure compliance with any applicable laws regarding electronic signatures and electronic transactions.

## II. Function Descriptions

This section describes the functions that are available with the eSign Signature Capture plug-in. All functions, except where noted, return a text string. It is up to the user to convert the returned value when necessary.

### **PCES\_CaptureSignature( BoundData )**

#### **Description:**

Displays a dialog on the monitor showing that the device is ready to capture a signature. The returned string contains the biometric signature information encrypted with the 'BoundData' data. This string is used to later render the signature.

#### **Parameters:**

*BoundData* = A string of variable length. This data will be bound to the captured signature.

#### **Return Values:**

The function will return either an error or an ASCII character string representing the signature.

### **PCES\_GetOperatingMode**

#### **Description:**

Returns a string representing the current operating mode of the plug-in.

#### **Parameters:**

None

#### **Return Values:**

'UNREGISTERED', 'DEMO' or 'LIVE'

### **PCES\_Register( ServerName ; ServerPort ; ServerPage ; licenseID )**

#### **Description:**

Registers the plug-in with Productive Computing's registration servers. Must call Register function once before using the plug-in.

#### **Parameters:**

ServerName = "licensing.productivecomputing.com"  
ServerPort = "80"  
ServerPage = "/PCIReg/pcireg.php"  
LicenseID = "your License ID" or "DEMO-ES1"

**Return Values:** 0 for success, otherwise an error

### **PCES\_RenderSigFromString( SigString ; BoundData )**

#### **Description:**

This function is used to render the signature that was captured using the signature capturing function. The return value is either a bitmap image of the string or the text 'UNSIGNED.' The return values should be stored in FileMaker 'Container' fields.

#### **Parameters:**

*SignatureASCII* = The string that contains the encrypted biometric signature information.

*BoundData* = The string containing the data that SigString is bound to.

#### **Return Values:**

Either an error, a bitmap rendering of the signature, or the text 'UNSIGNED'

#### **Note:**

The returned bitmap's dimensions are set at 1500x500 pixels.

### **PCES\_SaveSigToFile( SigString ; BoundData ; Path )**

#### **Description:**

This function saves the bitmap of the SigString signature bound to the BoundData string to the provided Path.

#### **Parameters:**

*SignatureASCII* = The string that contains the encrypted biometric signature information.

*BoundData* = The string containing the data that SigString is bound to.

*Path* = A valid Windows path. This also includes the desired file name.

#### **Return Values:**

0 for success, otherwise an error

#### **Note:**

Path must include the desired file name.

#### **Examples:**

```
PCES_SaveSigToFile( SomeTable::SigData ; SomeTable::SomeText ; "C:\\Users\\SomeUser\\Desktop\\Signature.bmp" )
```

### **PCES\_SetBannerPath( BMPLocation )**

#### **Description:**

This function changes the path for the signature banner bitmap to be displayed on the LCD signature tablet.

#### **Parameters:**

*Path* = A valid Windows path to the signature banner .BMP file

#### **Return Values:**

0 for success, otherwise an error

### **PCES\_SetImagePenWidth( Pixels )**

#### **Description:**

This function is used to set the width of the pen used to draw the signature in the bitmap returned by the signature rendering function.

#### **Parameters:**

*Pixels* = A number representing the width of the pen in pixels

#### **Return Values:**

0 for success, otherwise an error

#### **Note:**

The returned bitmap's dimensions are set at 1500x500 pixels.

### **PCES\_SetTabletComPort( PortNum )**

#### **Description:**

This function sets the signature tablet's COM port number to PortNum.

#### **Parameters:**

*PortNum* = An integer number representing the COM port that the signature tablet will communicate through.

#### **Return Values:**

0 for success, otherwise an error

#### **Note:**

Tablet Mod and Tablet COM Port are used in conjunction with the "BSB" model signature tablet device.

### **PCES\_SetTabletMode( Mode )**

#### **Description:**

This function sets the tablet mode to the provided Mode.

#### **Parameters:**

*Mode* = An integer representing the mode of the signature tablet.

#### **Return Values:**

0 for success, otherwise an error

#### **Note:**

Tablet Mod and Tablet COM Port are used in conjunction with the "BSB" model signature tablet device.

### **PCES\_Version( type )**

#### **Description:**

Used to identify the product and product version. The current version of the plug-in in string format. A 'Short' version contains only the version number. A 'Long' version contains the product name and the version number.

#### **Parameters:**

type = "Long" or "Short"

The short version string includes only the version number of the plug-in. Example: "1.0.3.0".

The long includes the name of the plug-in and version number. Example: "eSign Signature 1.0.3.0"

The default version string returned is long.

#### **Returns:**

Product name and/or version, otherwise an error.



### III. Return/Error Codes

Most functions return a text string. It is up to the user to convert the returned value when necessary. In the event that an error is returned, then use the following table for a list of error codes. Since all errors have a value that is less than zero, trapping for return values that are less than zero will trap only errors.

<b>Value</b>	<b>Meaning</b>
0	Success
-1	Plug-in not registered or session expired
-3	Invalid number of parameters
-10	Failed registration
-13000	Unknown file or access denied
-13001	Unable to save signature image to file
-13002	Ensure that the SigPlus software is properly installed

## IV. Contact Us

Successful integration of a FileMaker plug-in requires the creation of integration scripts within your FileMaker solution. A working knowledge of FileMaker Pro, especially in the areas of scripting and calculations is necessary. If you need additional support for scripting, customization or setup (excluding registration) after reviewing the videos, documentation, FileMaker demo and sample scripts, then please contact us via the avenues listed below.

Phone: 760-510-1200

Email: [support@productivecomputing.com](mailto:support@productivecomputing.com)

Forum: [www.productivecomputing.com/forum](http://www.productivecomputing.com/forum)

Please note assisting you with implementing this plug-in (excluding registration) is billable at our standard hourly rate. We bill on a time and materials basis billing only for the time in minutes it takes to assist you. We will be happy to create your integration scripts for you and can provide you with a free estimate if you fill out a Request For Quote (RFQ) at [www.productivecomputing.com/rfq](http://www.productivecomputing.com/rfq). We are ready to assist and look forward to hearing from you!