



Integration Guide

pDoc Pro Server SDK

Copyright © Topaz Systems Inc. All rights reserved.

For Topaz Systems, Inc. trademarks and patents, visit www.topazsystems.com/legal.

Table of Contents

1.0 – Introduction.....	4
2.0 – Installing/Uninstalling pDoc Pro Server SDK.....	4
2.1 – Before You Begin	4
2.2 – Installing pDoc Pro Server SDK	5
2.3 – Uninstalling pDoc Pro Server SDK	5
3.0 – Features of the SDK	5
4.0 – Add pDoc Pro Server SDK Reference to Your Project	6
5.0 – Namespaces.....	9
6.0 – Enumerations.....	10
6.1 The ClientStatus enumeration	10
6.2 The TransactionType enumeration	10
6.3 The HomeScreenType enumeration	10
7.0 – Classes.....	11
7.1 ClientInfo Class	11
7.1.1 Properties	11
7.2 ClientHomeScreen Class	12
7.2.1 Properties	12

Table of Contents

7.3 SigningClient Class	13
7.3.1 Properties	13
7.3.2 Methods	13
7.3.2.1 GetListOfClients.....	13
7.3.2.2 SelectClient.....	14
7.3.2.3 GetClientStatus.....	14
7.3.2.4 PushPDF	15
7.3.2.5 SetupHomeScreen	15
7.3.2.6 TerminateHomeScreen.....	16
7.3.2.7 TerminateSigningSession.....	17
7.3.3 Events	18
7.3.3.1 docReceived	18
8.0 Parameter File	20
8.1 Parameter File Schema (From Server to Client).....	20
8.2 Parameter File Schema (From Client to Server)	25

1.0 – Introduction

The pDoc® Pro eSign Software System provides an electronic signature solution for environments where a system operator pushes a PDF document from their PC to a client for signing. The pushed document can be a form, contract, or other document for a user to complete and sign. The user completes a document by filling in text boxes, selecting check boxes and radio buttons, selecting items from lists, and signing the document. When the user has finished and presses a done button icon, the client sends the completed document back to the originator's PC where it is stored and available to the operator. Then the client goes back to sleep, playing a video or displaying a sequence of images, until it receives another document for signing.

The pDoc Signer application running on the pDoc Pro Client captures the handwritten biometric signature plus the time and date of the captured signature. Also, the integrity value (hash) of the document will be calculated and stored in the signature field together with the biometric signature. The captured signature is displayed on the corresponding signature field in the document and can be viewed and verified by other PDF document rendering applications like Adobe Acrobat or Adobe Reader.

The pDoc Pro Server SDK is an SDK that can be integrated into applications that want to leverage the functionality of pDoc Pro. An application running on a PC can integrate this SDK and operate or manipulate the pDoc Pro Client.

2.0 – Installing/Uninstalling pDoc Pro Server SDK

The pDoc Pro Server SDK is part of pDoc Pro Server software.

2.1 – Before You Begin

- pDoc Pro Server runs on Windows operating systems beginning with Windows 7.
- Your system should have a minimum of 50 MB free space on the hard drive, in addition to the free space requirements for Windows.
- Before installing a new version of pDoc Pro Server, uninstall any older versions of pDoc Pro Server on your PC.
- pDoc Pro Clients may be dedicated or non-dedicated Windows tablets or PCs. See the separate pDoc Pro Client User Manual for additional information.

2.2 – Installing pDoc Pro Server SDK

The pDoc Pro Server SDK is installed as part of the pDoc Pro Server software. Installing pDoc Pro Server software is accomplished by running the pDoc Pro Server installation file provided. During the installation you will be provided with an option to agree or disagree to the license agreement, prompted to enter user information, and optionally set a different folder for installation. The sequence of screens displayed guide the user through the installation process.

2.3 – Uninstalling pDoc Pro Server SDK

The pDoc Pro Server SDK is uninstalled when the pDoc Pro Server software is uninstalled from a PC. To uninstall pDoc Pro Server:

- Go to Start → Control Panel → Programs and Features.
- Select pDoc Pro Server and click on the “Uninstall” button. Follow the instructions to uninstall the software.

The instructions are similar for other Windows operating systems.

Note: Deleting the pDoc Pro Server installation folder directly will not uninstall the software completely.

3.0 – Features of the SDK

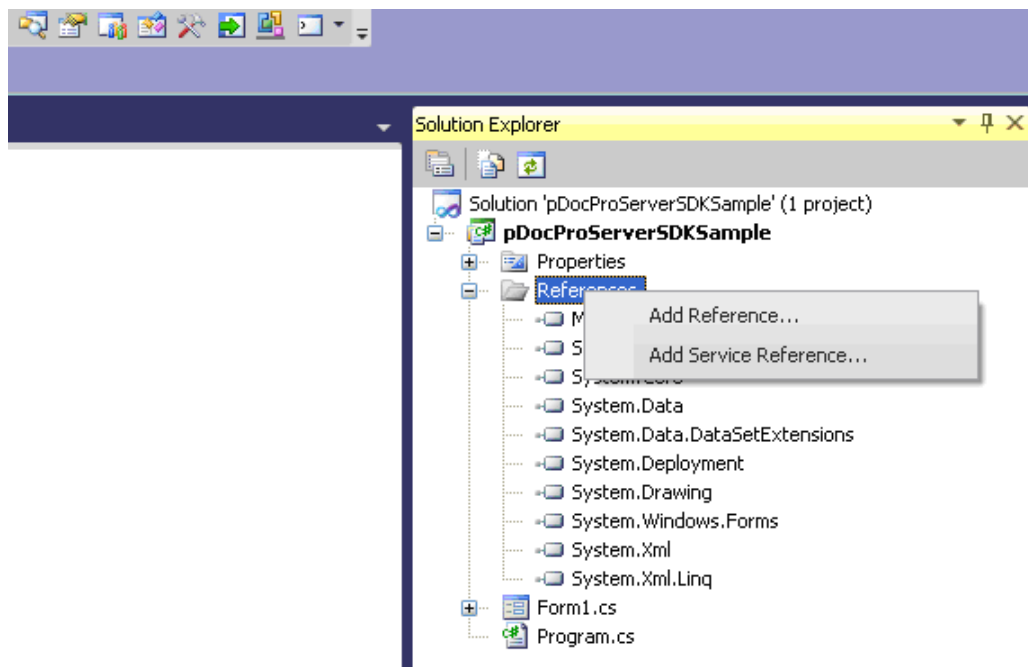
- Set the Client Information. Required client information is Client Identifier and the Port Number for communication with the client
- Check the client state (available, busy, not available)
- Terminate a signing session (e.g., in case of a timeout or other error).
- Select the PDF file to be pushed to the client.
- Select the parameter file to be pushed to the client.
- Push the PDF and associated parameter file to the specified client.
- Set the home screen display type (Basic, Images (default), or Video).
- Select the list of images to display on the home screen (maximum of 10)

- Set the time interval to display each image
- Select the video to be run
- Specify whether Video Player Bar should be displayed or not
- Setup the Home Screen (i.e., push the video or images to the client)
- Terminate the home screen on the client, i.e. revert it back to a normal Windows client
- Check if signed PDF document has been received from the client
- Get the signed PDF document received from the client
- Get the updated parameter file received from the client

4.0 – Add pDoc Pro Server SDK Reference to Your Project

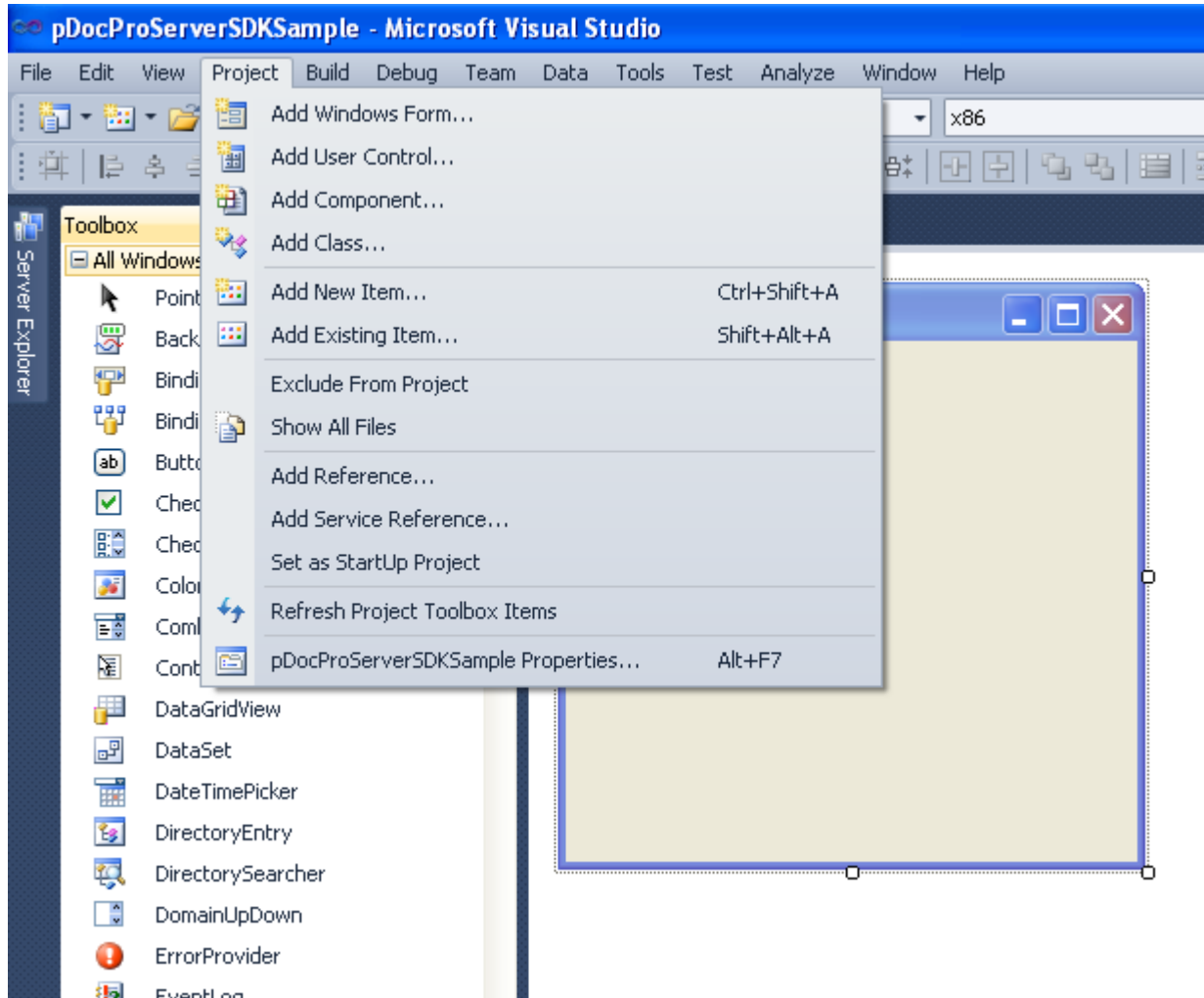
Before you begin to use the various API calls described in this document you should add a reference to the pDoc Pro Server SDK assembly to your project. This section describes how this can be done using Visual Studio 2008.

1. In the solution explorer right click on the project node and select “Add Reference”.

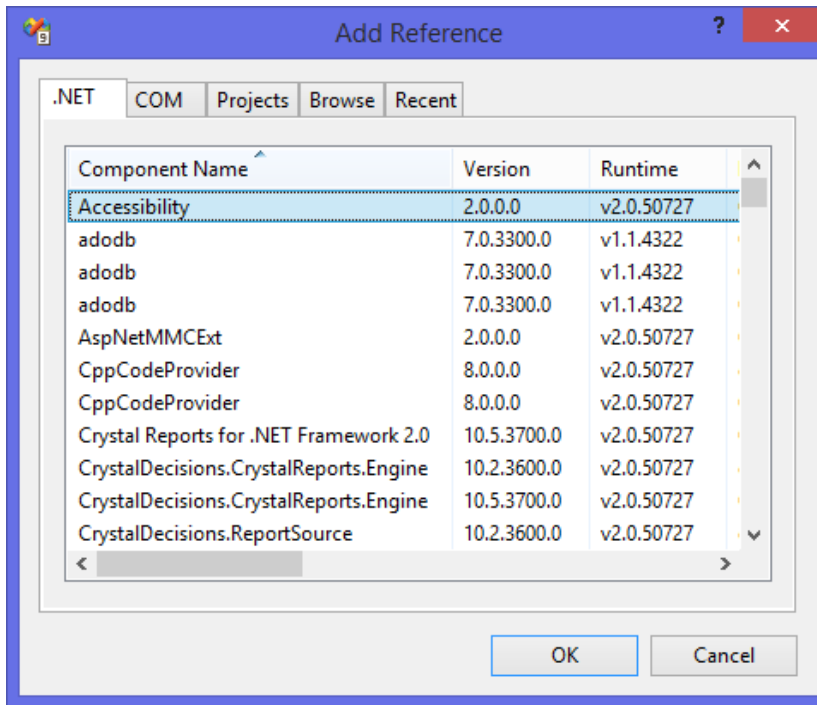


You can also add the reference using the main menu.

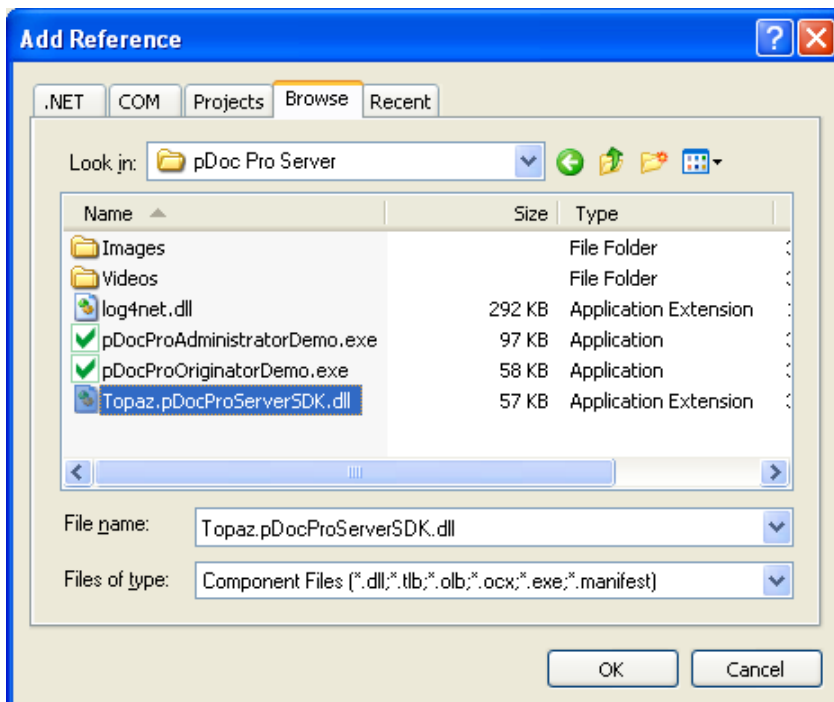
Navigate to “Project” and then select “Add Reference”.



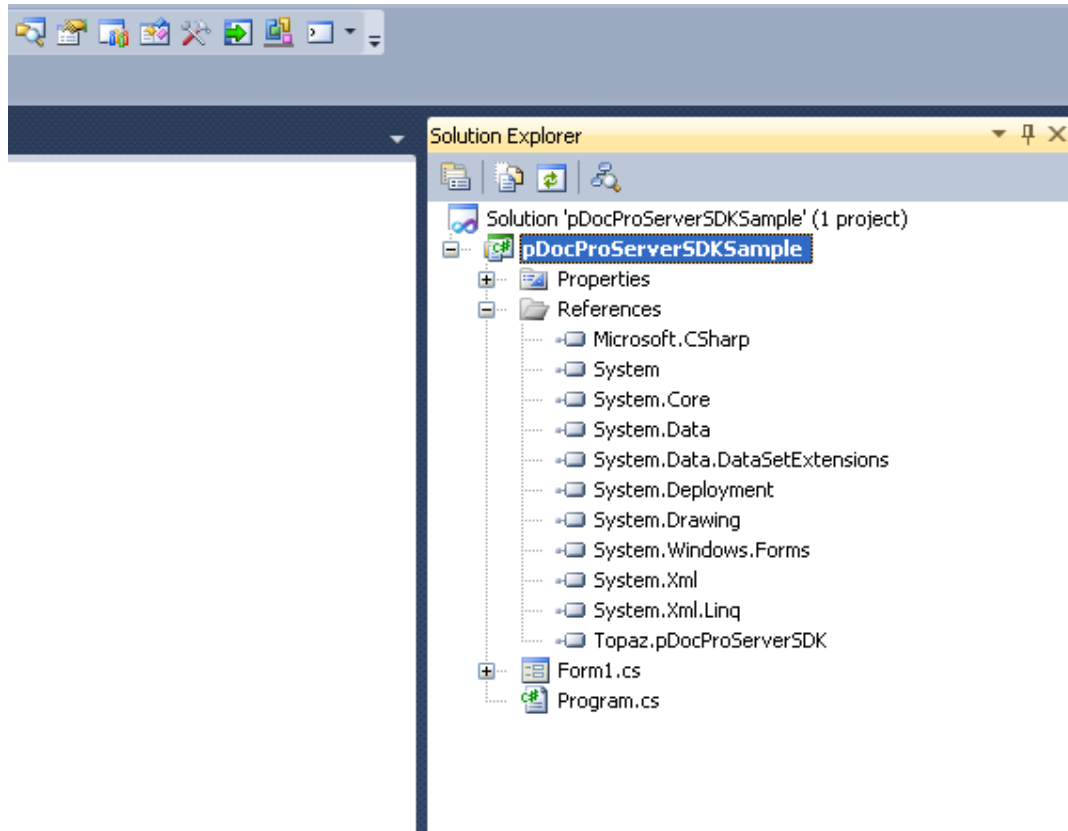
2. The following dialog will appear.



Select the “Browse” tab and navigate to the folder where the “Topaz.pDocProServerSDK.dll” is located (typically the pDoc Pro Server installation folder) and select the assembly. Click “OK” to add the assembly to your project.



3. The assembly will be added to the project as shown. Now you can start using the API exposed by the pDoc Pro Server SDK.



5.0 – Namespaces

The API calls are encapsulated within the following namespaces.

- Topaz.pDocProServerSDK.ClientInfo
 - This namespace holds the class containing information about a pDoc Pro Client.
- Topaz.pDocProServerSDK.ClientHomeScreen
 - This namespace holds the class containing information about a pDoc Pro Client Home Screen.
- Topaz.pDocProServerSDK.SigningClient
 - This namespace will hold the main class required for administering a pDoc Pro Client. This class will encapsulate methods, properties, and fields required to administer the pDoc Pro Client.

6.0 – Enumerations

6.1 The ClientStatus enumeration

Enumeration that lists the available client statuses.

Namespace

Topaz.pDocProServerSDK.SigningClient

Syntax

```
public enum ClientStatus
{
    Available = 1,
    Busy = 2,
    NotAvailable = 3
}
```

6.2 The TransactionType enumeration

Enumeration that lists the TransactionTypes.

Namespace

Topaz.pDocProServerSDK.SigningClient

Syntax

```
public enum TransactionType
{
    HomeScreenSetup = 1,
    PushPDF = 2,
    TerminateSigningSession = 3,
    UpdateLicenseKey = 4,
    TerminateHomeScreen = 5
}
```

6.3 The HomeScreenType enumeration

Enumeration that lists the HomeScreenTypes.

Namespace

Topaz.pDocProServerSDK.ClientHomeScreen

Syntax






```
public enum HomeScreenType
{
    Basic = 0,
    Images = 1,
    Video = 2
}
```

7.0 – Classes

7.1 ClientInfo Class

Class that provides information about a specific pDoc Pro Client. Contains information like Identifier (Name or IP Address) and Port Number used to communicate with the client.







7.1.1 Properties

	Name	Syntax	Description
	ClientIdentifier	<code>public string ClientIdentifier</code> { get; set; }	The identifier for the pDoc Pro Client. The identifier can be the name or IP Address of the client.
	PortNumber	<code>public int PortNumber</code> { get; set; }	The port number used to communicate with the pDoc Pro Client.
	ClientStatus	<code>public int ClientStatus</code> { get; set; }	The status of the client (Available, Not Available or Busy)
	SigningSessionTimeout	<code>public int SigningSessionTimeout</code> { get; set; }	Specifies the time out period for the signing session in minutes. Default value is 10 minutes.
	CheckForExternalPad	<code>public bool CheckForExternalPad</code> { get; set; }	Specifies whether signing with an External Signature Pad is allowed or not. Useful when the clients are Windows tablets to improve the signing experience. Default value is False.

7.2 ClientHomeScreen Class

Class that contains information about the home screen of the client.





7.2.1 Properties

	Name	Syntax	Description
	ImageDuration	<code>public int ImageDuration { get; set; }</code>	Time period to display each image in seconds
	ImageList	<code>public string[] ImageList { get; set; }</code>	List of images to be run on home screen
	VideoFile	<code>public string VideoFile { get; set; }</code>	Video to be run on the home screen
	Password	<code>public string Password { get; set; }</code>	Password to terminate the program on the pDoc Pro Client
	ScreenType	<code>public HomeScreenType screenType { get; set; }</code>	Home Screen Type (i.e., Basic, Images, or Video)
	DisplayVideoPlayerBar	<code>public bool DisplayVideoPlayerBar { get; set; }</code>	Specifies if the video player bar needs to be displayed or not. Applicable only when the home screen type is set to "Video"

7.3 SigningClient Class

The primary class of the SDK that an application can use to administer a pDoc Pro Client.

7.3.1 Properties

	Name	Syntax	Description
	PDFFilePath	<code>Public string PDFFilePath</code> <code>{ get; set; }</code>	The path of the PDF File to be pushed to the client.
	ParameterFilePath	<code>Public string</code> <code>ParameterFilePath</code> <code>{ get; set; }</code>	The path of the Parameter File to be pushed to the client.
	SendingTime	<code>Public string SendingTime</code> <code>{ set; }</code>	The time at which the PDF document was sent to the client.
	ErrorCode	<code>Public string ErrorCode</code> <code>{ set; }</code>	Error Code returned if any of the functions is not successful.

7.3.2 Methods

7.3.2.1 GetListOfClients

Method Description

Provides the configured list of clients (by the administrator) as an array of ClientInfo classes.

Syntax

```
public ClientInfo[] GetListOfClients(string AuthID)
```

Parameters

AuthID
Type: string
Authentication Identifier.

Return Value

ClientInfo array

7.3.2.2 *SelectClient*

Method Description

Allows selecting a pDoc Pro Client with which to start communication.

Syntax

```
public void SelectClient(ClientInfo clientInfo)
```

Parameters

clientInfo

Type: ClientInfo

ClientInfo class containing the Name or IP Address, and Port Number of the pDoc Pro Client with which to communicate.

Return Value

No return value

7.3.2.3 *GetClientStatus*

Method Description

Gets the status of the specified pDoc Pro Client.

Syntax

```
public ClientStatus GetClientStatus(string AuthID)
```

Parameters

AuthID

Type: string

Authentication Identifier.

Return Value

ClientStatus containing the pDoc Pro Client status (Available, Busy, and NotAvailable).

7.3.2.4 PushPDF

Method Description

Pushes the selected PDF to the specified pDoc Pro Client. The PDFFilePath and ParameterFilePath properties have to be set before this method is invoked.

Syntax

```
public bool PushPDF(string AuthID)
```

Parameters

AuthID
Type: string
Authentication Identifier.

Return Value

Boolean indicating the status of sending the PDF to the client for signing.
If PushPDF is not successful, the return value is false and the application should query the ErrorCode property to get the error code. The following error codes are applicable.

Error Code	Description
ERRDATACONSTRUCTION	An error occurred while preparing the data to be sent to the Client.
ERRSTARTCONNECTION	A connection with the client could not be established.
ERRMSGNULL	No response received from the client.
ERRAUTHFAILED	The Authentication ID provided is incorrect.
ERRCLIENTBUSY	The client is busy.
ANOTHERTRANSRUNNING	Another transaction is running on the client.
PUSHPDFFAIL	Push PDF transaction failed because of an exception. Check the logs for details.

7.3.2.5 SetupHomeScreen

Method Description

Sets up the home screen of the specified pDoc Pro Client with the information provided.

Syntax

```
public bool SetupHomeScreen(ClientHomeScreen clientScreen, string AuthID)
```

Parameters

clientScreen
Type: ClientHomeScreen
Client Home Screen information

AuthID
Type: string
Authentication Identifier.

Return Value

Boolean indicating whether the home screen setup was successful or not.

If SetupHomeScreen is not successful, the return value is false and the application should query the ErrorCode property to get the error code. The following error codes are applicable.

Error Code	Description
ERRDATACONSTRUCTION	An error occurred while preparing the data to be sent to the Client.
ERRSTARTCONNECTION	A connection with the client could not be established.
ERRMSGNULL	No response received from the client.
ERRAUTHFAILED	The Authentication ID provided is incorrect.
ERRCLIENTBUSY	The client is busy.
ANOTHERTRANSRUNNING	Another transaction is running on the client.
NOTSUPPORTED	Setup home screen transaction is not supported by the selected client.
SETUPHOMESCREENFAIL	Setup home screen transaction failed because of an exception. Check the logs for details.

7.3.2.6 TerminateHomeScreen

Method Description

Terminates the home screen of the client, regardless of its current state, and returns the client to normal usage.

Syntax

```
public bool TerminateHomeScreen(string AuthID)
```


Parameters

AuthID

Type: string

Authentication Identifier.

Return Value

Boolean indicating whether the home screen was terminated successfully or not.
Returns true if the home screen is not currently running on the client.

If TerminateHomeScreen is not successful, the return value is false and the application should query the ErrorCode property to get the error code. The following error codes are applicable.

Error Code	Description
ERRDATACONSTRUCTION	An error occurred while preparing the data to be sent to the Client.
ERRSTARTCONNECTION	A connection with the client could not be established.
ERRMSGNULL	No response received from the client.
ERRAUTHFAILED	The Authentication ID provided is incorrect.
ERRCLIENTBUSY	The client is busy.
ANOTHERTRANSRUNNING	Another transaction is running on the client.
NOTSUPPORTED	Terminate home screen transaction is not supported by the selected client.
TERMINATEHOMESCREENFAIL	Terminate Home Screen transaction failed because of an exception. Check the logs for details.

7.3.2.7 TerminateSigningSession

Method Description

Terminates the signing session in progress, resets to the home screen, and waits for the next signing session to be initiated.

Syntax

```
public bool TerminateSigningSession(string AuthID)
```

Parameters

AuthID
Type: string
Authentication Identifier.

Return Value

Boolean indicating whether the signing session was terminated successfully or not. Returns true if the signing session is not currently active on the client.

If TerminateSigningSession is not successful, the return value is false and the application should query the ErrorCode property to get the error code. The following error codes are applicable.

Error Code	Description
ERRDATACONSTRUCTION	An error occurred while preparing the data to be sent to the Client.
ERRSTARTCONNECTION	A connection with the client could not be established.
ERRMSGNULL	No response received from the client.
ERRAUTHFAILED	The Authentication ID provided is incorrect.
ERRCLIENTBUSY	The client is busy.
ANOTHERTRANSRUNNING	Another transaction is running on the client.
TERMINATESIGNINGSESSIONFAIL	Terminate Signing Session transaction failed because of an exception. Check the logs for details.

7.3.3 Events

7.3.3.1 docReceived

Event Description

This event is raised when the document is received back from the pDoc Pro Client after completion of signing process in the client. The DocEventArgs will contain the PDF File Name, Parameter File Name, etc.

Syntax

```
void SigningClient_docReceived(object sender,  
Topaz.pDocProServerSDK.SigningClient.DocReceivedArgs DocArgs)
```

Parameters

DocReceivedArgs contain the following information.

clientIdentifier
portNumber
sentPDFFileName,
sentTime,
receivedPDFFilePath
receivedParameterFilePath
receivedTime
docSignedStatus
receivedMessage

clientIdentifier – The client identifier from which the document was received

portNumber – Port Number from which the document was received

sentPDFFileName – The name of the PDF that was sent to the client

sentTime – The time at which the PDF was sent to the client

receivedPDFFilePath – The path of the processed PDF that was received and saved by the SDK. The PDF is received back from the client only when the signer clicks the “Done” button in the client either after signing or without signing. In all other cases like Cancelled, Terminated, Error Occurred, License Expired or Timed Out, the document is not returned back and this parameter would be empty.

receivedParameterFilePath– The path of the updated parameter file that was received and saved by the SDK. The updated parameter file is received back from the client only when the signer clicks the “Done” button in the client either after signing or without signing. In all other cases like Cancelled, Terminated, Error Occurred, License Expired or Timed Out, the updated parameter file is not returned back and this parameter would be empty

receivedTime – The time at which the response was received from the client.

docSignedStatus – This is a Boolean value that indicates if the document was signed during the signing session or the document was returned back without any signatures being signed in that session. A value of “1” indicates that the document was signed in the session, and a value of “0” indicates that the document was not signed in the session.

receivedMessage – Contains the return message from the client. The value can be one of the following

- SIGNSUCCESS
- SIGNCANCELLED
- SIGNERROR
- PDOCEXPIRED
- SIGNINGTERMINATED
- SIGNINGTIMEDOUT
- COMMUNICATIONERROR

8.0 Parameter File

When the system operator pushes a PDF document to a client machine, the server software also sends a parameter file to the client along with the PDF document. The parameter file controls the pDoc Pro Client behavior on the client.

The parameter file contains the information configured by the Administrator using the Administrator Demo module. The default parameter file, named DefaultParam.xml, is installed by the pDoc Pro Server installation into the Common Application Data folder of the server machine. Typically this folder is "C:\ProgramData\pDoc Pro Server". Custom applications can manipulate this file directly for writing any other information required to control the pDoc Pro Client behavior. When the parameter file is pushed to the client, it should have the same name as the PDF that is currently being pushed. Section 8.1 below explains the schema of the parameter file along with a sample file.

When the signed PDF is received by the system operator, along with the PDF document, another variant of the parameter file is also received that contains the information about the signing session and also the sequence of events happened during the signing session. Section 8.2 explains the schema of this file along with a sample file.

8.1 Parameter File Schema (From Server to Client)

```
<?xml version="1.0" encoding="utf-8"?>
<!--Root element for the Parameter File-->
<pDocProServer SchemaVersion="1.0">
  <!--Name of the PDF file-->
  <PDFName />
  <!--Size of the PDF file-->
  <PDFSize />
  <!--Specified timeout period for Signing-->
  <SigningTimeOut />
```

```
<!--Specified whether Pen Display extended monitor is allowed or not-->
<EnablePenDisplay />
<!--Signature information like First Name, Last Name and Email can be added here if known-->
<SignatureInformation>
  <FirstName />
  <LastName />
  <Email />
</SignatureInformation>
<!-- Signature Capture Related information-->
<SignatureCaptureData>
  <!--Specified whether signing with external signature capture devices is allowed or not -->
  <ExternalDevice />
  <!--Signature Position with respect to details(i.e. behind, beside and Image Only) -->
  <SignPosition />
  <!--Specified if Maximum Enlargement Feature needs to be enabled (TRUE or FALSE)-->
  <EnableMEF>
    <!--Maximum Enlargement Factor percentage-->
    <MEFPercent />
  </EnableMEF>
  <!--Inking color for the signature-->
  <InkColor />
</SignatureCaptureData>
<!--Attributes for the Text Comments feature. Specifies whether text comments feature is enabled or
not and if enabled, different attributes for font-->
<TextComments Enabled="">
  <FSize />
  <FStyle />
  <FEffct />
  <FColor />
  <FName />
```

</TextComments>

<!--Attributes for the Pencil markup feature. Specifies whether pencil markup feature is enabled or not. If enabled, the line color and line width for the pencil markups.-->

<PencilMarkups Enabled="">

<LineColor />

<LineWidth />

</PencilMarkups>

<!--Specifies if the signature field list window is enabled or not (TRUE or FALSE)-->

<EnableSignFieldList />

<!--Specifies if the attachment list window is enabled or not (TRUE or FALSE)-->

<EnableAttachmentList />

<!--Text to be displayed on the pDoc Signing Screen user interface before clicking on a signature field-->

<DisplayTextBeforeSign />

<!--Text to be displayed on the pDoc Signing Screen user interface after clicking on a signature field-->

<DisplayTextSign></DisplayTextSign>

<!--"Done" button options-->

<DoneButton>

<!--Display text for the done button-->

<DisplayText></DisplayText>

<!--Message that needs to be displayed when the done button is clicked-->

<MessageText>MessageText</MessageText>

<!--Text for the "Cancel" button in the message-->

<CancelButTextMsg> </CancelButTextMsg>

<!--Text for the "Continue" button in the message-->

<ContinueButTextMsg> </ContinueButTextMsg>

</DoneButton>

<!--"Cancel" button options-->

<CancelButton>

<!--Display text for the Cancel button-->

```
<DisplayText></DisplayText>
<!--Message that needs to be displayed when the Cancel button is clicked-->
<MessageText></MessageText>
<!--Text for the "Yes" button in the message-->
<YesButtonTextMsg></YesButtonTextMsg>
<!--Text for the "No" button in the message-->
<NoButtonTextMsg></NoButtonTextMsg>
</CancelButton>
</pDocProServer>
```

The following is a sample parameter file. The name of the PDF that is pushed in this example is pDoc Pro Sample Document.pdf and hence the parameter file name should be pDoc Pro Sample Document.xml.

```
<?xml version="1.0" encoding="utf-16"?>
<pDocProServer SchemaVersion="1.0">
  <PDFName>pDoc Pro Sample Document.pdf</PDFName>
  <PDFSize>651047</PDFSize>
  <SignatureInformation>
    <FirstName>
    </FirstName>
    <LastName>
    </LastName>
    <Email>
    </Email>
    <SigningTimeOut> 10 </SigningTimeOut>
    <EnablePenDisplay> True </EnablePenDisplay>
  </SignatureInformation>
  <SignatureCaptureData>
    <ExternalDevice>True</ExternalDevice>
```

```

<SignPosition>Image Only</SignPosition>

<EnableMEF>

  <MEFPercent>25</MEFPercent>

</EnableMEF>

<InkColor>0000ff</InkColor>

</SignatureCaptureData>

<TextComments Enabled="false">

  <FSize>26</FSize>

  <FStyle>Regular</FStyle>

  <FEffect>False</FEffect>

  <FColor>#000000</FColor>

  <FName>Times New Roman</FName>

</TextComments>

<PencilMarkups Enabled="false">

  <LineColor>#000000</LineColor>

  <LineWidth>1</LineWidth>

</PencilMarkups>

<EnableSignFieldList>False</EnableSignFieldList>

<EnableAttachmentList>False</EnableAttachmentList>

<DisplayTextBeforeSign>To sign, select a signature area below.</DisplayTextBeforeSign>

<DisplayTextSign>Sign your name in the signing window.</DisplayTextSign>

<DoneButton>

  <DisplayText>DONE</DisplayText>

  <MessageText>Do you agree that the information you have provided is complete and correct, and
that you want to conduct business electronically?</MessageText>

  <CancelButTextMsg>I do not agree</CancelButTextMsg>

  <ContinueButTextMsg>I agree</ContinueButTextMsg>

</DoneButton>

<CancelButton>

  <DisplayText>CANCEL</DisplayText>

```


`<MessageText>`This action will delete any signatures and data you have entered. Do you want to continue this cancel process?`</MessageText>`

`<YesButtonTextMsg>`Yes`</YesButtonTextMsg>`

`<NoButtonTextMsg>`No`</NoButtonTextMsg>`

`</CancelButton>`

`</pDocProServer>`

8.2 Parameter File Schema (From Client to Server)

`<?xml version="1.0" encoding="utf-16"?>`

`<!--Root element for the Parameter File-->`

`<pDocProServer SchemaVersion="1.0">`

`<!--Name of the PDF file-->`

`<PDFName />`

`<!--Size of the PDF file-->`

`<PDFSize />`

`<!--Pages visited during the signing session -->`

`<PagesVisited />`

`<!--Whether the document was signed during this signing session or not. 0 – Not Signed and 1 - Signed-->`

`<DocumentSigned/>`

`<!--Sequence of events during the signing session -->`

`<Audit>`

`<Record1/>`

`<Record2/>`

`</Audit>`

`</pDocProServer>`

The following is a sample parameter file. The name of the PDF that is pushed in this example is Introducing pDoc Signer.pdf and hence the parameter file name should be Introducing pDoc Signer-S1.xml.

```
<?xml version="1.0" encoding="utf-16"?>
<pDocProServer SchemaVersion="1.0">
  <PDFName>Introducing pDoc Signer.pdf</PDFName>
  <PDFSize>601394</PDFSize>
  <PagesVisited>1,2,3,4,5,6,7,8,9</PagesVisited>
  <DocumentSigned>1</DocumentSigned>
  <Audit>
    <Record1>2~2~10/26/15 02:05:09 PM</Record1>
    <Record2>1~2~1~1~2</Record2>
    <Record3>1~3~1~2~3</Record3>
    <Record4>1~4~1~3~4</Record4>
    <Record5>1~4~10~100~75</Record5>
    <Record6>1~4~9~75~100</Record6>
    <Record7>1~5~1~4~5</Record7>
    <Record8>1~6~1~5~6</Record8>
    <Record9>1~6~2~Favorite Color~ ~Green</Record9>
    <Record10>1~6~3~Eastern~Eastern~Pacific</Record10>
    <Record11>1~7~1~6~7</Record11>
    <Record12>1~7~6~Sales~Off~ales</Record12>
    <Record13>1~6~1~7~6</Record13>
    <Record14>1~6~2~Favorite Color~Green~Red</Record14>
    <Record15>1~7~1~6~7</Record15>
    <Record16>1~7~5~States~Alabama~California</Record16>
    <Record17>1~7~5~Multiple Selection List Box~High School~High
School,Bachelor's,Doctorate</Record17>
    <Record18>1~8~1~7~8</Record18>
    <Record19>1~8~4~Loan~Auto~Home Improvement</Record19>
    <Record20>1~9~1~8~9</Record20>
    <Record21>1~9~7~sample Annotation~sample Annotation Text</Record21>
    <Record22>1~9~6~Maternal Grandmother~Off~MGrandma</Record22>
```

```
<Record23>1~9~6~Maternal Grandpa~Off~MGrandpa</Record23>
<Record24>1~9~5~List Box~Internet Search~Internet Search,Newspaper Ad</Record24>
<Record25>1~9~8~Patient~10/26/15 02:06:50 PM~ </Record25>
<Record26>2~1~10/26/15 02:06:52 PM</Record26>
</Audit>
</pDocProServer>
```

The following events are recorded during the signing session:

Page Level Events:

1. Page Navigation
2. Text Box Change
3. Radio Button Selection Change
4. Combo Selection Change
5. List Selection Change
6. Check Box Selection Change
7. Text Annotation Change
8. Signature Field Signed
9. Zoom In
10. Zoom Out

Document Level Events

1. Done
2. Open

Below is the format of the event under the <Audit> tag for each of the above events.

1. Page Navigation

1 ~ Page Number ~ 1 ~ Current Page ~ New Page

2. Text Box Change

1 ~ Page Number ~ 2 ~Field Name~ Previous Value ~ New Value

3. Radio Button Selection Change

1 ~ Page Number ~ 3 ~Field Name~ Previous Export Value ~ New Export Value

4. Combo Selection Change

1 ~ Page Number ~ 4 ~Field Name~ Previous Value ~ New Value

5. List Selection Change

1 ~ Page Number ~ 5 ~Field Name~ Previous Value(s) with comma as seperator ~ New Value(s) with comma as seperator

6. Check Box Selection Change

1 ~ Page Number ~ 6 ~Field Name~ Previous Export Value(s) ~ New Export Value(s)

7. Text Annotation Change

1 ~ Page Number ~ 7 ~Field Name~ Previous Value ~ New Value

8. Signature Field Signed

1 ~ Page Number ~ 8 ~ Signature Field Name ~ Date Time ~ Signer Name

9. Zoom In

1 ~ Page Number ~ 9 ~ Current Zoom Percentage ~ New Zoom percentage

10. Zoom Out

1 ~ Page Number ~ 10 ~ Current Zoom Percentage ~ New Zoom percentage

Document Level Events

1. Done

2 ~ 1 ~ Date Time at which the “Done” button was clicked

2.Open

2~2~Date Time at which the Document was opened in the pDoc Pro Client