



XPRESSDOX SERVER

Additional Information

(Updated May 2016)

PRIVATE & CONFIDENTIAL

The enclosed material is private and confidential. It is supplied for the purpose of product information and may not be disclosed in any manner other than to the addressee, the addressee's employees or authorised representatives. XpressDox is not responsible for errors or omissions contained in this document and nothing herein shall be construed as binding upon XpressDox. All rights and obligations between the parties shall be subject to mutually agreed upon contractual documentation. XpressDox reserves the right to modify the enclosed and/or all related details at any time thereafter.

CONTENTS

1. WHAT IS XPRESSDOX SERVER	3
2. ON-PREMISE SERVER	4
3. CLOUD SERVER	5
4. XPRESSDOX API	6
5. WORD DESKTOP	9
6. SALESFORCE INTEGRATION	10
7. XPRESSDOX CLOUD SERVER – SECURITY	11
8. REQUIREMENTS FOR ON-PREMISE SERVER	13
9. INTEGRATE THE INTERVIEW INTO YOUR WEBSITE	14
10. JAVASCRIPT INTEGRATION	14
11. IFRAME INTEGRATION	16
12. RESTful API AVAILABLE	18
13. RELATED LINKS	25

WHAT IS XPRESSDOX SERVER?

XpressDox Server is arguably the most powerful and flexible Cloud-hosted (we host) or On-premise (you host) document automation server available on the market today. It allows enterprises to implement cost-effective and flexible document automation solutions for use internally, or by external (client-facing) users.

Document templates may be run on Microsoft® Word or via any modern browser. XpressDox Server is also available as an Application Program Interface (API) which means that it can be integrated with an in-house application, or even your website.



ON-PREMISE SERVER

Secure, private and confidential document automation in your own scalable server environment with extensive deployment options



CLOUD SERVER

The quickest and easiest way to get up and running without compromising your security. User and template management is included as standard



WORD DESKTOP

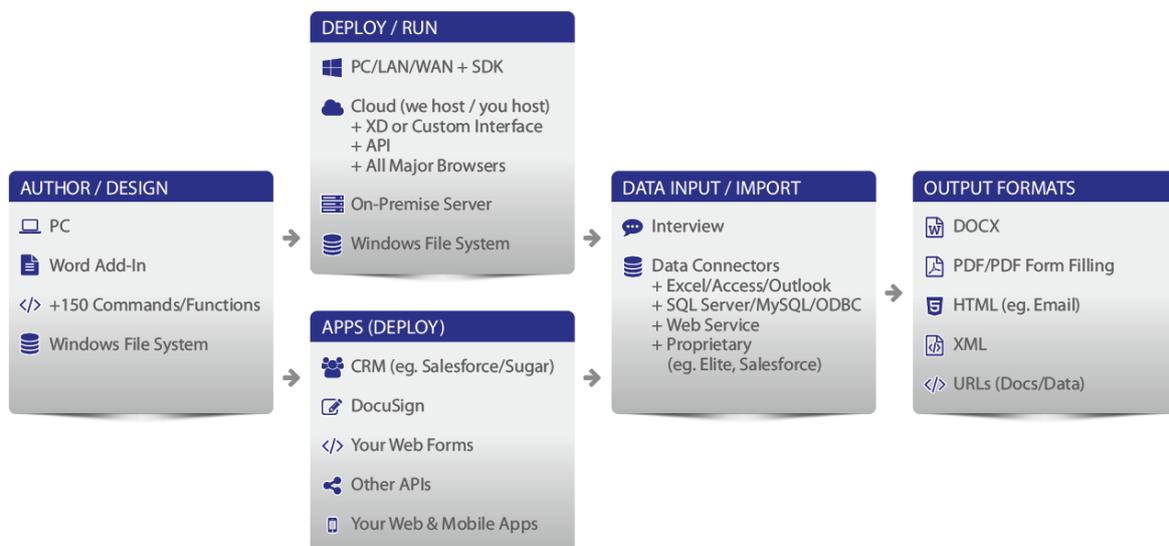
Design and run templates in Word. Powerful and flexible desktop document automation. Generates documents in Word, PDF and other formats



API

Develop apps with XpressDox API. Integrate the XpressDox template engine into your apps using APIs for Word, Web, .NET and COM

XPRESSDOX “SPACE”



ON-PREMISE SERVER

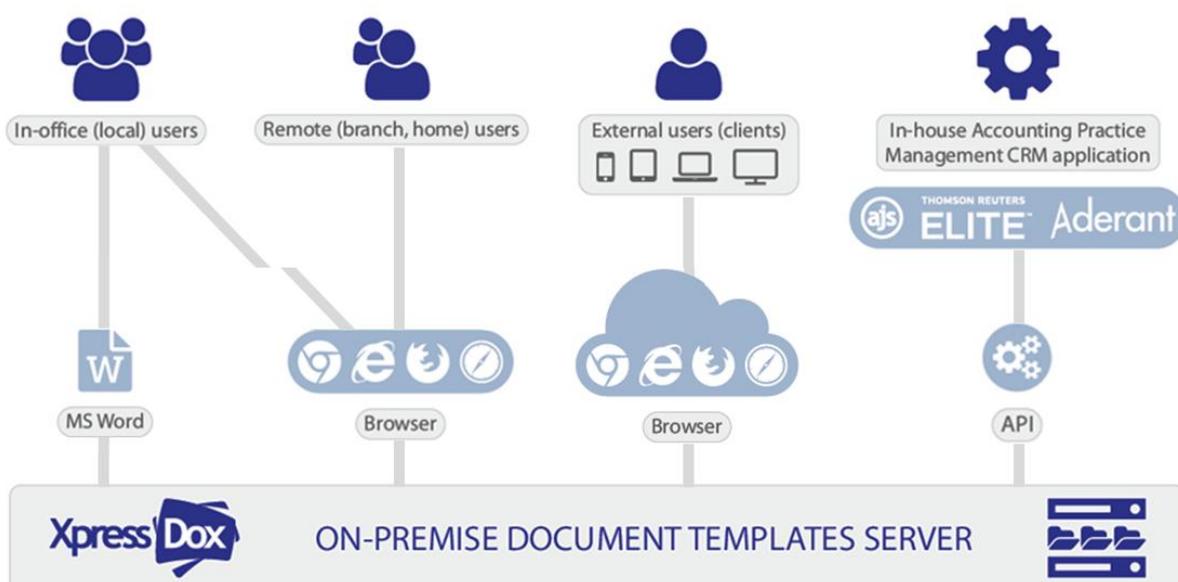
For those larger enterprises who prefer to host the XpressDox Server on their own servers (on-premise or hosted) XpressDox offers an unlimited-users licensing option on rental or outright purchase. Users run templates using the XpressDox Desktop version for Word, or any modern web-browser. Completed documents may be in Word and/or PDF format. Template authoring is done in Microsoft® Word for Windows.

Users access the same template regardless of whether they are using Microsoft® Word or a browser, which makes editing and updating of templates much simpler. Access to the system is managed by Windows Authentication, which means that you won't need to configure any user access rights, since XpressDox simply conforms to your existing security policies. This saves time and makes installation and administration simpler.

The XpressDox on-premise server is ideal where your document templates will access external data sources, for example an in-house CRM or accounting database. XpressDox server includes native data source compatibility, including SQL Server, ODBC-compliant databases, MySQL, Microsoft Office data sources such as Access, Excel, and Outlook, XML, text files and applications such as Salesforce.com.

XpressDox on-premise server is licensed per server, with additional test or development servers available at an 80% discount. Because XpressDox is fast, large organizations can often get by with only a single server, although some firms choose to have more than one server for redundancy or geographic reasons.

ON-PREMISE SERVER – USER ACCESS OPTIONS



CLOUD SERVER

For large organizations, where cost of software deployment is an issue, it makes sense to deploy templates via the web. Users run templates using any modern web-browser, and completed documents may be in Word and/or PDF format. Template authoring is done in Microsoft® Word for Windows.

Since templates are hosted in the Cloud, new templates are instantly available to all users regardless of location (branch office, home, or on the road), and version enhancements are available to all users instantly.

XpressDox Cloud also executes your routing instructions, so that completed documents will be automatically emailed to internal or external recipients specified in each template. It doesn't stop there. Your customers and clients can even enter routing information themselves.

Access to the Cloud is via a login and password, with point to point encryption over https, ensuring that your templates are completely private. Administrators can add and disable users, configure settings, and even change the theme or add a logo to the user interface. Users run templates from a folder structure which is similar in layout to Windows Explorer.

It is also possible to run your cloud-based templates from a link on your website, using the Include in your webpage function. XpressDox currently supports various integrations, including iFrame, JavaScript, or RESTful API integration.

XPRESSDOX API

Access XpressDox Document Automation functionality directly from your .NET, COM, Web, or Word applications, to produce more complete and powerful document-rich solutions.

Using the API a developer can write code that gathers data from any data base or other location, determine via program logic and interaction with users what templates are to be merged, and can call on the API to merge the data into one or more templates.

XPRESSDOX API FOR THE WEB

Two mechanisms are available to access XpressDox over the Internet. Using the hyperlink API, you would be able to upload selected templates onto the XpressDox server, into your own secure area. After that, you would add hyperlinks into you own web site which would enable a user to choose from the list of templates uploaded, and then run one or more of the templates. The interview which is then presented to them in the browser will be almost exactly the same as the desktop interview which XpressDox-in-Word presents to the Word user. After completing the interview, your web site can then contain a hyperlink which permits the user to download the merged document (if that is what you want) from the server to their local PC. In addition, the template author can specify an email address (or list of addresses) to which the merged document and the captured data set will be sent. This email address would typically be that of an internal staff member who would process the results in the appropriate way.

There is an implementation of the XpressDox Web Server which can be hosted on your company's intranet. This would enable all users who are registered to use your intranet to run XpressDox templates via a web browser (for example, Internet Explorer, Mozilla Firefox, Google Chrome or Safari). The look-and-feel of the XpressDox Web Server version is very much like that experienced when a user selects to "run a template" using the XpressDox Word Add-in. In other words, the user is presented with in interface which mimics the desktop XpressDox Explorer from which they select the template(s) to be run. The user is then presented with an interview, which again mimics the interview presented by the Word Add-in version, including functionality such as "Use Previous Data" or "Use Other Data", and access to data sources, etc. Once the merged document is created, it is stored in a location on the server which is configured using the normal XpressDox configuration mechanism. The same holds for the location of the captured data set. The user can then download the merged document from the server onto their local desktop, in either Word or PDF format, depending on how this is specified by the template author.

XPRESSDOX-IN-WORD API

Develop custom interfaces but with all the features of XpressDox Desktop. This API grants the developer access to most of the functionality provided for the Word Add-in version of XpressDox, which is denied to the APIs as noted above. The XpressDox-In-Word API would be used specifically

where there is a need to develop a custom user interface for the capture of data and/or the selection of templates, but where the developer would still like to have access to the features made available by the Word Add-in. Use of this API requires Microsoft® Office Word to be installed not only for rendering the merged documents, but for some of the other pre- and post-merging functions as well.

The XpressDox-In-Word API and documentation is installed along with the installation of the XpressDox Word Add-in. The documentation is provided in the Word document XpressDox In Word API Specification.doc which will have been installed into the My Documents\XpressDox folder the first time Word is loaded after XpressDox has been installed. The Specification document refers the developer to a Word template in the My Documents\XpressDox\Samples folder. This Word template contains some Word VBA macros which demonstrate the usage of some of the XpressDox-In-Word API functions as called from within a VBA macro.

XPRESSDOX API FOR .NET

Typically this API would be used to embed XpressDox document assembly capability into systems such as accounting, practice management, work flow and any other system where there is a need for creating non trivial documents. The templates used can be sourced from the Windows file system, or from any other source such as a document management system.

If there is no need to render the merged documents on the machine on which the API will operate, then it is not necessary to have Microsoft® Office Word installed on that system. Word is needed in order to author templates, which in turn requires XpressDox Desktop, and to render the resulting merged document for printing or reading. The function of merging the template and data that XpressDox performs does not require Word.

The XpressDox API is embedded in the .NET assemblies which are installed with the Word Add In instance of XpressDox, so no additional software is needed in order to use the API. But a sample Visual Studio solution (in C#) can be downloaded, along with formal documentation on the use of the API.

XPRESSDOX API FOR COM

The second XpressDox API is for use by non .NET developers and it exposes an interface to XpressDox via COM. This interface is very similar to the XpressDox Engine API for .NET, the major difference being that templates are expected to reside somewhere on the Windows file system. This does not mean that they cannot actually be stored in a document management system or database, but that they will have to be moved onto the file system before the COM API will be able to access them.

Similarly to the XpressDox API for .NET, the COM API uses assemblies that are installed along with the Word Add-in instance of XpressDox. It also needs another assembly, which is downloaded and installed along with the XpressDox API in the download process referred to above. The COM API would be used in much the same circumstances as the XpressDox API for .NET.

Using either of the two APIs, the developer is responsible for the creation of an XML data set and also the selection of the template and the storing of the merged document created from them. It is necessary to note that much of the functionality available to the XpressDox Word Add-in is NOT available to the API user. This includes the dynamic data capture interview and all the features exposed via the Configuration user interface (e.g. Standard Folders for document and data storage, Data Sources, Standard Data Items, Configuration merging, etc.). All commands which can be coded into templates which refer to these features (i.e. all the Data Capture commands in the Command Editor, as well as Data Source commands and commands in the Advanced document and file handling section of the CE) will be ignored when the template merging is handled via either of the two XpressDox Engine APIs. However, the XpressDox SDK, which is a new addition (and can be downloaded from the same place as the API download) will make all of the above exclusions available to the developer – try it and see.

WORD DESKTOP

XpressDox Desktop is an add-in to Microsoft® Office Word 2007 or newer. Templates and data are stored using Word XML to ensure compatibility with other systems. Completed documents may be generated in Word or PDF format for more portability.

XpressDox Desktop versions include Author, which is for template designers. XpressDox Author may also be used to run templates. For those users who do not need to design templates, there is the XpressDox Runner version. Lastly, XpressDox Supervisor is used by larger enterprises for configuring user access rights, data-sources, and file name and save locations for completed documents.

Extensive help is provided for template authors. To create a template, experienced authors can simply type commands and functions directly into a Word document. These commands and functions include those for every document automation task you may need.

Inexperienced template authors may find it easier to access the power of XpressDox through the Command Editor, which presents the syntax, as well as samples and wizards, for each command or function. Pressing F1 in the Command Editor accesses the extensive help on the web site.

Inexperienced template authors may find it easier to access the power of XpressDox through the Command Editor, which presents the syntax, as well as samples and wizards, for each command or function. Pressing F1 in the Command Editor accesses the extensive help on the web site.

When a template is run, an Interview is generated automatically. XpressDox allows template authors to make interviews as easy as possible for users to complete, with features like user help, captions, tabs, headings and footings, validation rules, and conditional display.

XpressDox Desktop includes native data source compatibility, including SQL Server, ODBC-compliant databases, MySQL, Microsoft Office data sources such as Access, Excel, and Outlook, XML, text files and applications such as Salesforce.com. In addition, when running templates you can choose to capture new data, re-use recently captured data, or use previously saved data.

XpressDox Desktop provides multi-language authoring including English (US and GB), and French. Template interviews may be created in any Western language.

SALESFORCE INTEGRATION

In today's competitive business environment your sales people can't wait for information. If they do you can bet your customers are waiting too. Here are 3 simple ways to eliminate waiting:

- **Store data in one place.** There should be no reason you have customer information replicated outside ofsalesforce.com. Once the data exists in salesforce.com it is available to users in Word via XpressDox so that letters, quotes, and contracts can be produced in seconds.
- **Capture all data.** Make sure all prospect and customer information is captured to salesforce.com. With a complete data set to choose from your sales people will not waste time.
- **Lessen data leakage risks.** With all prospect and customer information securely stored in salesforce.com, not on laptops and other mobile devices, intentional and unintentional data leakage is minimized.

AVOID RETYPING AND DUPLICATE RECORDS

Sales reps and managers are already working long, hard hours. They need technology to help them work smarter and more productively. Look here for quick productivity gains:

- **Avoid retyping information.** Retyping information already captured in salesforce.com is an absolute no-no. XpressDox can automatically include customer information in correspondence so no data needs to be retyped.
- **Merge duplicate records.** When prospecting plays a big part in filling the sales funnel, you end up with multiple records for a single prospect. Merge and discard duplicates relentlessly.

ENFORCE STANDARDS COMPLIANCE

Many companies still struggle with letterhead standardization with hundreds of letterheads in operation. Here's how XpressDox can help eliminate wasted time caused by wrong stationery:

- **One letterhead.** Use one letterhead so that brand, styles, and contact details are correct across your entire organization.
- **Customized contact details.** Ideal for multi-office and international companies, XpressDox creates the correct contact details and references for any document.
- **Repetitive tasks.** Set XpressDox to work by automating your repetitive tasks, like billing or new customer welcome letters for example.

XPRESSDOX CLOUD SERVER - SECURITY

TECHNICAL

Your account in the XpressDox Cloud is only accessible by you or by any users you create in that account via the relevant username and password combination. No one else has access to your documents or data. All communication between your browser and the XpressDox Cloud occurs over https and is fully encrypted. (If you would like to read more about SSL – please see this link: <https://www.digicert.com/ssl.htm>) Our servers are dedicated (only XpressDox has access to them) and they are hosted in a secure data center behind an industry-standard firewall. Note that for increased peace of mind you can elect to use the XpressDox “Windows Authentication” version of the XpressDox Server which would be installed locally on your network, and would be secured behind your own firewall and network security.

INTELLECTUAL PROPERTY RIGHTS

XpressDox acknowledges and agrees that (i) the Client will hold all intellectual property rights in its content, data and transmissions incorporated in the Service including, but not limited to, copyright and trademark rights, and (ii) XpressDox will hold all intellectual property rights in the tools and techniques incorporated in the Service including, but not limited to, copyright and trademark rights.

PRIVACY OF PERSONAL INFORMATION

XpressDox will comply with all reasonable requests from the Client with respect to protecting Personal Information of the Client's employees, customers and suppliers it receives in connection with its performance of this Agreement. To the extent that either Party processes Personal Information of the Client, including but not limited to Personal Information relating to its employees, customers and suppliers, each Party agrees that it will:

Restrict access to Personal Information to employees or agents who are properly authorised to process such Personal Information and who, by virtue of their office or contract are subject to appropriate confidentiality obligations;

Follow the Client's instructions in connection with Processing such Personal Information;

Implement reasonable, appropriate technical and organisational measures to preserve the integrity and confidentiality of the Personal Information and to prevent any unauthorised Processing, access, use, corruption or loss of the Personal Information;

Verify that all security measures that are in place are effectively implemented;

Not transfer or process Personal Information outside of South Africa to recipients that are not subject to appropriate data protection principles unless consent of the Client is obtained to do so;

Not disclose any such Personal Information to any third party without the prior written consent of the Client or unless required by law;

Conduct regular assessments to identify all reasonable foreseeable internal and external risks to the Personal Information in that Party's possession or control and update and align safeguards with the risks identified;

Delete any Personal Information in its possession or control upon the expiry of the applicable retention period as prescribed by law, or upon the expiry or termination of this Agreement, or within 10 days of a written request by the Client requesting the deletion or handing over of such Personal Information, whichever occurs first, unless otherwise agreed upon between the Parties; and

Provide the Client upon reasonable advance notice with access to its premises to ensure that appropriate security measures are in place to protect the Personal Information.

In protecting Personal Information as set out in clause 1 above, the Parties agree to adhere to all data privacy laws including but not limited to the Protection of Personal Information Act, 4 of 2013, and in the event of any unauthorised, unlawful and/or unintended Processing or where there are reasonable grounds to believe that the Personal Information has been accessed or acquired by any unauthorised person, each Party will immediately notify the Client and co-operate with all reasonable requests to investigate and remediate such incident and provide appropriate response.

REQUIREMENTS FOR ON-PREMISE SERVER

From a software perspective, the following is required in order to run the on-premise XpressDox server on an intranet using Windows Authentication:

1. Server requirements - Windows server (2008 or higher)
2. .Net Framework 4 is required as well as IIS7 or higher
3. Requires access to Windows network domain controller (Active Directory for authentication)
4. Users will be authenticated using their Windows network/machine login credentials

The server exposes the files and folders configured via the Office Settings section of the XpressDox configuration, which allows the Supervisor to configure site-wide settings that apply across the entire network, for instance the ability to configure and force the save location of all documents and data files for any given template.

BENEFITS INCLUDE:

1. All Windows network file/folder access rights are automatically adhered to – no need to re-configure this in XpressDox
2. No need to roll out and install the desktop application on all users' computers, a browser is used to run the templates
3. Templates can be run remotely via the Internet and the documents will automatically be saved to the configured location on your network
4. Ease of access to all existing databases and other applications on the network

INTEGRATE THE INTERVIEW INTO YOUR WEBSITE

From the XpressDox Cloud Explorer Interface you can click to the right of a template name and then select "Include in your Page". This will provide the following information to help you to integrate XpressDox Interviews in your website. Note that the template identifiers will obviously change for each template depending on name and location.

JAVASCRIPT INTEGRATION

Important to Note: Although this method offers a more seamless integration, JavaScript and CSS conflicts can occur between the XpressDox code and your own code.

- 1) Copy the below script tags and paste them at the top of the **<head>** tag of your code page.

```
<!-- If you are already using the jQuery JavaScript Library in your page, do not include this line (See note above) -->
<script type="text/javascript" src="https://server.xpressdox.com/cloud/Integration/LoadScript.ashx?id=jquery"></script>
<!-- This JavaScript file is required to load the XpressDox interview as well as the code required to run it -->
<script type="text/javascript" src="https://server.xpressdox.com/cloud/Integration/LoadScript.ashx?id=xpressdoxLoader"></script>
```

- 2) Copy the below HTML code and paste it at the desired location within the **<body>** tag of your code page.

```
<!-- This value identifies the template - the template can be moved or renamed without affecting this identifier -->
<input id="xdTemplateIdentifier" name="xdTemplateIdentifier" value="832e8178-795a-4360-b907-b2301f03f87d" type="hidden">
<!-- This value is only applicable when using the RESTful API - see last tab above -->
<input id="xdUsageIdentifier" name="xdUsageIdentifier" value="" type="hidden">
<!-- This value identifies the user profile -->
<input id="xdUserIdentifier" name="xdUserIdentifier" value="63ac13fd-11f4-476a-8c5c-c37218bf1da8" type="hidden">

<!-- This value can be set to empty if you choose to use the default XpressDox results page -->
<input id="xdReturnURL" name="xdReturnURL" value="" type="hidden">

<!-- You can pre-populated the interview with data by setting this value. Initial data must be in XML format consistent with that
found in XpressDox data files -->
<!-- Important - the xml data must be encoded to Base64 format for the post to be successful -->
<input id="xdInitialData" name="xdInitialData" value="" type="hidden">

<!-- The interview will be loaded into this container -->
<div id="divInterviewContainer">
  <!-- This HTML will display a loading gif while the interview and code are loading from the XpressDox cloud -->
  <div id="divInterviewLoading" style="height: 500px;">
    <div style="margin: 100px auto 0px auto; height: 64px; width: 64px; background:
url('https://server.xpressdox.com/Cloud/ComponentStyling/LoaderMask/Images/Loader.gif') no-repeat;"></div>
    <div style="margin: 15px auto; height: 20px; text-align: center; font-weight: bold;">Loading...</div>
  </div>
</div>

<!-- Possible values for ThemeName are: Default, Albatross, Cardinal, ElectricOrange, FiscalShrike, Lorie, LumoGreen, PowderBlue,
PurpleTurtle,RoseRed, SlateBlue, Turaco, Wagtail -->
<input id="xdthemeName" name="xdthemeName" value="Default" type="hidden">

<!-- Possible values for InterviewSize are: Small, Medium, Large and xLarge -->
```

```
<input id="xdInterviewSize" name="xdInterviewSize" value="Medium" type="hidden">
```

- 3) If you have chosen to use a custom result/completion page, **XML** in the following format will be posted (HTTP POST) via an HTML form to the specified URL on completion of the interview. The name of the parameter containing the xml data will be set to **xdResultData**.

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfAssemblyUsage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <AssemblyUsage>
    <Identifier>9a4238c7-d241-4b2a-afb3-b0b2fd69a2e0</Identifier>
    <TemplatePath>Demonstration Folder\ExampleTemplate.xdtpx</TemplatePath>
    <TemplateIdentifier>84dc3a91-64db-473b-95a5-3eb27f55c747</TemplateIdentifier>
    <DataFilePath>Demonstration Folder\Data\ExampleData.xddata.xml</DataFilePath>
    <DataDownloadLink>
      https://server.xpressdox.com/Cloud/Integration/Download.aspx?usageIdentifier=9a7938c7-d208-4b2a-afb3-b0b2fd69a2f0
    </DataDownloadLink>
    <DataFileSize>569 bytes</DataFileSize>
    <DataFileDateCreated>2012-07-25T14:41:43.7755426Z</DataFileDateCreated>
    <AssembledFiles>
      <AssembledFile>
        <Identifier>42b3afda-f175-4366-9a4b-e5e6fc6a7a48</Identifier>
        <FilePath>Demonstration Folder\Documents\ExampleDocument.docx</FilePath>
        <DownloadLink>
          https://server.xpressdox.com/Cloud/Integration/Download.aspx?fileIdentifier=42b3ghda-f175-4166-9a4b-e5e6fc6a5g48
        </DownloadLink>
        <FileType>MS Word 2007</FileType>
        <FileSize>20 KB</FileSize>
        <DateCreated>2012-07-25T14:41:43.7615418Z</DateCreated>
      </AssembledFile>
      <AssembledFile>
        <Identifier>42b3afda-f175-4166-9a6n-e5e6fc6a7a48</Identifier>
        <FilePath>Demonstration Folder\Documents\ExampleDocument.pdf</FilePath>
        <DownloadLink>
          https://server.xpressdox.com/Cloud/Integration/Download.aspx?fileIdentifier=42b3afda-f175-4166-9a4b-e5e6fc6a7a48
        </DownloadLink>
        <FileType>PDF Document</FileType>
        <FileSize>34 KB</FileSize>
        <DateCreated>2012-07-25T14:41:43.7615418Z</DateCreated>
      </AssembledFile>
    </AssembledFiles>
  </AssemblyUsage>
</ArrayOfAssemblyUsage>
```

IFRAME INTEGRATION

Important to Note: iFrame integration is the simplest of the 3 options to implement.

- 1) To display an interview in your page, simply copy the below HTML code and paste it at the desired location within the **<body>** tag of your code page.

```
<!-- This HTML will create an iFrame on your page in which the interview will be displayed -->

<iframe id="xpressDoxInterview" frameborder="0" style="width:100%; height:600px;"
src="https://server.xpressdox.com/Cloud/Integration/LoadInterview.aspx?templateIdentifier=832e8178-795a-4360-b907-
b2301f03f87d&userIdentifier=63ac13fd-11f4-476a-8c5c-c37218bf1da8&returnURL=&size=Large&theme=Default"></iframe>

<!-- Documentation of URL Parameters -->

<!-- templateIdentifier value identifies the template - the template can be moved or renamed without affecting this identifier -->
<!-- userIdentifier value identifies the user profile -->
<!-- returnUrl - this value can be set to empty if you choose to use the default XpressDox results page -->
<!-- Possible values for theme are: Default, Albatross, Cardinal, ElectricOrange, FiscalShrike, Lourie, LumoGreen, PowderBlue,
PurpleTurtle,RoseRed, SlateBlue, Turaco, Wagtail -->
<!-- Possible values for size are: Small, Medium, Large and xLarge -->

<!-- Any additional parameter you choose to include will also be included in the returnUrl for the return post -->
```

- 2) To pre-populate the interview with data will require a little more work than the approach discussed above.

- a. Create a new page in your website/project called **PostXpressDoxData.html** and paste the below code into the **<body>** tag of this code page. Please note you will need to replace the text **XML_DATA_IN_BASE64_FORMAT** with your Base64 encoded XML. This value can also be set programmatically.

```
<!-- An HTML form is used to post (HTTP POST) the data to the XpressDox interview URL -->

<form id="xpressDoxForm" method="post"
action="https://server.xpressdox.com/Cloud/Integration/LoadInterview.aspx?templateIdentifier=832e8178-795a-4360-b907-
b2301f03f87d&userIdentifier=63ac13fd-11f4-476a-8c5c-c37218bf1da8&returnURL=&size=Large&theme=Default">

  <!-- Initial data must be in XML format consistent with that found in XpressDox data files -->
  <!-- Important - the xml data must be encoded to Base64 format for the post to be successful -->
  <input name="initialData" value="XML_DATA_IN_BASE64_FORMAT" type="hidden" />
</form>

<!-- This script will submit the above HTML form on load of your page -->
<script type="text/javascript">
  window.onload = function() {
    var xdForm = document.getElementById("xpressDoxForm");
    xdForm.submit();
  }
</script>
```

- b. Create a new page in your website/project and paste the below code into the **<body>** tag of this code page.

```
<!-- This HTML will create an iFrame on your page in which the interview will be displayed -->

<iframe id="xpressDoxInterview" frameborder="0" style="width:100%; height:600px;" src="PostXpressDoxData.html"></iframe>
```

3. If you have chosen to use a custom result/completion page, **XML** in the following format, will be posted (HTTP POST) via an HTML form to the specified URL on completion of the interview. The name of the parameter containing the xml data will be set to **xdResultData**.

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfAssemblyUsage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <AssemblyUsage>
    <Identifier>9a4238c7-d241-4b2a-afb3-b0b2fd69a2e0</Identifier>
    <TemplatePath>Demonstration Folder\ExampleTemplate.xdtpx</TemplatePath>
    <TemplateIdentifier>84dc3a91-64db-473b-95a5-3eb27f55c747</TemplateIdentifier>
    <DataFilePath>Demonstration Folder\Data\ExampleData.xdata.xml</DataFilePath>
    <DataDownloadLink>
      https://server.xpressdox.com/Cloud/Integration/Download.aspx?usageIdentifier=9a7938c7-d208-4b2a-afb3-b0b2fd69a2f0
    </DataDownloadLink>
    <DataFileSize>569 bytes</DataFileSize>
    <DataFileDateCreated>2012-07-25T14:41:43.7755426Z</DataFileDateCreated>
    <AssembledFiles>
      <AssembledFile>
        <Identifier>42b3afda-f175-4366-9a4b-e5e6fc6a7a48</Identifier>
        <FilePath>Demonstration Folder\Documents\ExampleDocument.docx</FilePath>
        <DownloadLink>
          https://server.xpressdox.com/Cloud/Integration/Download.aspx?fileIdentifier=42b3ghda-f175-4166-9a4b-e5e6fc6a5g48
        </DownloadLink>
        <FileType>MS Word 2007</FileType>
        <FileSize>20 KB</FileSize>
        <DateCreated>2012-07-25T14:41:43.7615418Z</DateCreated>
      </AssembledFile>
      <AssembledFile>
        <Identifier>42b3afda-f175-4166-9a6n-e5e6fc6a7a48</Identifier>
        <FilePath>Demonstration Folder\Documents\ExampleDocument.pdf</FilePath>
        <DownloadLink>
          https://server.xpressdox.com/Cloud/Integration/Download.aspx?fileIdentifier=42b3afda-f175-4166-9a4b-e5e6fc6a7a48
        </DownloadLink>
        <FileType>PDF Document</FileType>
        <FileSize>34 KB</FileSize>
        <DateCreated>2012-07-25T14:41:43.7615418Z</DateCreated>
      </AssembledFile>
    </AssembledFiles>
  </AssemblyUsage>
</ArrayOfAssemblyUsage>
```

RESTful API AVAILABLE

Important to Note: Using the RESTful API (Server to Server implementation) provides the Integrater with the most power and control, although it does require some development know-how.

The API offers the following functionality:

- Login to your XpressDox Cloud account
- Browse your XpressDox templates and associated files that have been uploaded/assembled to/in your account
- Retrieve an interview for a specified template and display it within your web application
- Assemble a specified template by posting XML data - with or without displaying an interview
- Downloading of assembled documents and data files – the documents and data can be easily imported directly into your application.

XPRESSDOX CLOUD – RESTful API

This document contains the technical specification of the XpressDox API used by third party vendors to integrate the power of XpressDox Document Assembly into existing applications.

It is a RESTful API and all response data is returned in XML format. Example data is provided below.

If you do not already have an account, click [here](#) to register an XpressDox Cloud account.

API Methods:

1. Login:

Used to log in to the client's XpressDox Cloud account.

[GET]

<https://server.xpressdox.com/Cloud/IntegrationServices/ThirdPartyService.svc/Login?username={username}&password={password}>

Returns an authenticationTicket (string) to be used in subsequent requests.

Response XML

```
<?xml version="1.0"?>
<string>
    3dd5363d-a32c-4cb9-92ca-250d5841bfea
</string>
```

2. Explore Folder:

Used to retrieve the contents of a folder.

Note: Send an empty string for folderIdentifier to retrieve the contents of the root folder.

[GET]

<https://server.xpressdox.com/Cloud/IntegrationServices/ThirdPartyService.svc/ExploreFolder?authenticationTicket={authenticationTicket}&folderIdentifier={folderIdentifier}>

Returns a list of ServerFile elements.

Response XML

```
<?xml version="1.0"?>
<ArrayOfServerFile>
  <ServerFile>
    <DateUploaded>2011-07-21T13:21:42.1257334Z</DateUploaded>
    <Description/>
    <FileSize/>
    <FileType>Folder</FileType>
    <Identifier>ZHy3+GEK9kTdM8VW2fi20XG740R7Wra</Identifier>
    <Name>Client Templates</Name>
    <TemplateIdentifier></TemplateIdentifier>
  </ServerFile>
  <ServerFile>
    <DateUploaded>2011-08-03T12:31:50.0878272Z</DateUploaded>
    <Description/>
    <FileSize>123 KB</FileSize>
    <FileType>XpressDox Template</FileType>
    <Identifier>GNYS988N3joKWhCzAY8uRkleOUOcfMibhazeR8UeOYw=</Identifier>
    <Name>contact-information.xdtp</Name>
    <TemplateIdentifier>4e01c495-911f-4c69-9d02-6dea2a58912b</TemplateIdentifier>
  </ServerFile>
</ArrayOfServerFile>
```

3. Get Template URL using File Identifier:

Used to retrieve the URL of the web interview.

The template is identified by the **fileIdentifier**, this identifier is dynamic and may change across sessions.

The template URL returned is static and may be used over and over again. Ideal for use in a static web page where the user may assemble as many document as they wish.

(For more information regarding the interview integration, please see the related links at the end of this document)

[GET]

<https://server.xpressdox.com/Cloud/IntegrationServices/ThirdPartyService.svc/GetTemplateURL?authenticationTicket={authenticationTicket}&fileIdentifier={fileIdentifier}&returnURL={returnURL}>

Returns a URL to be used in an iFrame within the third party application in order to display the interview.

Response XML

```
<?xml version="1.0"?>
<string>
  https://server.xpressdox.com/cloud/Integration/LoadInterview.aspx?templateIDs=0ef5408
  e-13a6-4dfe-a048-
  41bd7cebd641&redirectURL=https://www.YourDomain.com/DocumentAssembled
</string>
```

4. Get Template URL using Template Identifier:

Used to retrieve the URL of the web interview. Similar to above method.

The template is identified by the **templateIdentifier**, this identifier is static, it is assigned the first time the template is uploaded to the cloud and remains the same regardless whether the template is edited, renamed or moved.

The template URL returned can only be used for a single assembly. Ideal for use in dynamic web applications or ecommerce sites where assemblies are sold.

(For more information regarding the interview integration, please see the related links at the end of this document)

[GET]

```
https://server.xpressdox.com/Cloud/IntegrationServices/ThirdPartyService.svc/GetTemplateURLByTemplateI
D?authenticationTicket={authenticationTicket}&templateIdentifier={templateIdentifier}&returnURL={return
URL}
```

Returns a URL to be used in an iFrame within the third party application in order to display the interview.

Response XML

```
<?xml version="1.0"?>
<string>
  https://server.xpressdox.com/cloud/Integration/LoadInterview.aspx?templateIDs=0ef5408
  e-13a6-4dfe-a048-
  41bd7cebd641&redirectURL=https://www.YourDomain.com/DocumentAssembled
</string>
```

5. Get Single Use Template URL using File Identifier:

Used to retrieve the URL of the web interview.

The template is identified by the **fileIdentifier**, this identifier is dynamic and may change across sessions.

The template URL returned can only be used for a single assembly. Ideal for use in dynamic web applications where the user can browse the list of templates and then choose to run a particular interview.

(For more information regarding the interview integration, please see the related links at the end of this document)

[GET]

<https://server.xpressdox.com/Cloud/IntegrationServices/ThirdPartyService.svc/GetSingleUsageTemplateURL?authenticationTicket={authenticationTicket}&fileIdentifier={fileIdentifier}&returnURL={returnURL}>

Returns a URL to be used in an iFrame within the third party application in order to display the interview.

Response XML

```
<?xml version="1.0"?>
<string>
  https://server.xpressdox.com/cloud/Integration/LoadInterview.aspx?templateIDs=0ef5408
  e-13a6-4dfe-a048-
  41bd7cebd641&redirectURL=https://www.YourDomain.com/DocumentAssembled
</string>
```

6. Get Single Use Template URL using Template Identifier:

Used to retrieve the URL of the web interview. Similar to above method.

The template is identified by the **templateIdentifier**, this identifier is static, it is assigned the first time the template is uploaded to the cloud and remains the same regardless whether the template is edited, renamed or moved.

The template URL returned can only be used for a single assembly. Ideal for use in dynamic web applications or ecommerce sites where assemblies are sold.

(For more information regarding the interview integration, please see the related links at the end of this document)

[GET]

<https://server.xpressdox.com/Cloud/IntegrationServices/ThirdPartyService.svc/GetSingleUsageTemplateURLByTemplateID?authenticationTicket={authenticationTicket}&templateIdentifier={templateIdentifier}&returnURL={returnURL}>

Returns a URL to be used in an iFrame within the third party application in order to display the interview.

Response XML

```
<?xml version="1.0"?>
<string>
  https://server.xpressdox.com/cloud/Integration/LoadInterview.aspx?templateIDs=0ef5408
  e-13a6-4dfe-a048-
  41bd7cebd641&redirectURL=https://www.YourDomain.com/DocumentAssembled
</string>
```

7. Fill Template using File Identifier:

Used to assemble a specified template on the server.

The template is identified by the **fileIdentifier**, this identifier is dynamic and may change across sessions.

Data used to fill the document is sent in XML format, please see below for details.

[POST]

<https://server.xpressdox.com/Cloud/IntegrationServices/ThirdPartyService.svc/FillTemplate?authenticationTicket={authenticationTicket}&fileIdentifier={fileIdentifier}>

Returns the list of assembled documents, containing a download link for each document.

Post Data

```
<FaxCoverExample >
  <firstName>Nick</firstName>
  <lastName>Prince</lastName>
  <companyName>XpressDox (PTY) Ltd</companyName>
  <jobTitle>Developer</jobTitle>
  <email>nickp@o2smart.com</email>
  <web>server.xpressdox.com</web>
  <business>0214569770</business>
  <mobile>0834152702</mobile>
  <fax>0214568972</fax>
</FaxCoverExample >
```

Response XML

```
<?xml version="1.0"?>
<ArrayOfAssembledDocument >
  <AssembledDocument>
    <DownloadLink>https://server.xpressdox.com/cloud/Integration//Download.aspx?usageID=4a342405-ec6e-4c76-b53a-ff39a0073da4&fileType=doc
    </DownloadLink>
    <Name>contact-information.doc</Name>
  </AssembledDocument>
</ArrayOfAssembledDocument>
```

8. Fill Template using Template Identifier:

Used to assemble a specified template on the server. Similar to above method.

The template is identified by the **templateIdentifier**, this identifier is static, it is assigned the first time the template is uploaded to the cloud and remains the same regardless whether the template is edited, renamed or moved.

Data used to fill the document is sent in XML format, please see below for details.

[POST]

<https://server.xpressdox.com/Cloud/IntegrationServices/ThirdPartyService.svc/FillTemplateByTemplateID?authenticationTicket={authenticationTicket}&templateIdentifier={fileIdentifier}>

Returns the list of assembled documents, containing a download link for each document.

Post Data

```
<FaxCoverExample >
  <firstName>Nick</firstName>
  <lastName>Prince</lastName>
  <companyName>XpressDox (PTY) Ltd</companyName>
  <jobTitle>Developer</jobTitle>
  <email>nickp@o2smart.com</email>
  <web>server.xpressdox.com</web>
  <business>0214569770</business>
  <mobile>0834152702</mobile>
  <fax>0214568972</fax>
</FaxCoverExample >
```

Response XML

```
<?xml version="1.0"?>
<ArrayOfAssembledDocument >
  <AssembledDocument>
    <DownloadLink>https://server.xpressdox.com/cloud/Integration/Download.aspx?usageID=4a342405-ec6e-4c76-b53a-ff39a0073da4&fileType=doc
    </DownloadLink>
    <Name>contact-information.doc</Name>
  </AssembledDocument>
</ArrayOfAssembledDocument>
```

9. Upload File:

Used to upload a file or template to the server.

The folder to upload file to is identified by the **folderIdentifier**.

File to be uploaded is posted multipart form-data.

Post Data

[multipart/form-data] Please see: <http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4>

[POST]

<https://server.xpressdox.com/Cloud/Integration/upload.ashx?authenticationTicket={authenticationTicket}&olderIdentifier={folderIdentifier}>

Returns a ServerFile element.

Response XML

```
<?xml version="1.0" encoding="utf-8"?>
<ServerFile>
  <DateUploaded>2011-08-03T12:31:50.0878272Z</DateUploaded>
  <Description/>
  <FileSize>123 KB</FileSize>
  <FileType>XpressDox Template</FileType>
  <Identifier>GNYS988N3joKWhCzAY8uRkleOUOcfMibhazeR8UeOYw=</Identifier>
  <Name>contact-information.xdtpl</Name>
  <TemplateIdentifier>4e01c495-911f-4c69-9d02-6dea2a58912b</TemplateIdentifier>
</ServerFile>
```

10. Download File:

Used to download a file or template from the server.

The file or template is identified by the **downloadIdentifier**.

[GET]

<https://server.xpressdox.com/Cloud/Integration/download.aspx?authenticationTicket={authenticationTicket}&downloadIdentifier={downloadIdentifier}>

Return the file as a download.

RELATED LINKS:

Integration Information - For more information about integrating an interview into your web page, please see the documentation built into XpressDox Cloud. To access the integration information, login to [XpressDox Cloud](#), click to view any interview and then click the “[Include Interview in your Page](#)” link on the left had side of the page (link also available in the dropdown menu on the explorer page). Answer the very simple form and hit the “Show Me How” button and the code you need, along with an explanation, will be generated for you.

Template Commands – The following commands can be used in the template for further customization.

[SetWebReturnURL](#)

[SetWebButtonText](#)

[SendWebEmail](#)

[SetWebInterimSaveButton](#)

[SetWebInterviewSize](#)

(Click the above links for further information)