



# DocuPhase SDK / API

## *Developer's Guide*

Supported Versions: 6.1, 6.2

Last Revised: February 13, 2018

Copyright © 2018, iDatix Corporation dba DocuPhase Corporation. All rights reserved.

### DISCLAIMER

DocuPhase has made every effort to ensure that this document is accurate;  
DocuPhase disclaims liability for any inaccuracies or omissions.

No part of the contents of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form without written consent from DocuPhase Corporation.

Information in this document is subject to change without notice and does not represent commitment on the part of DocuPhase.

If you find information in this document that is incorrect, misleading, or incomplete, we would appreciate your comments and suggestions.

## Table of Contents

<b>Purpose of this Guide</b> .....	<b>1</b>
<b>DocuPhase SDK Overview</b> .....	<b>1</b>
SDK/API System Overview .....	2
<b>Application Command Line Arguments</b> .....	<b>3</b>
Basic Command Line Arguments Used for Both ScanDox and Barcoder .....	3
Working with ScanDox .....	5
Working with Barcoder .....	6
<b>DocuPhase SpeedLoader</b> .....	<b>7</b>
Working with SpeedLoader .....	7
<b>Setting Up and Using the DocuPhase URL API</b> .....	<b>9</b>
System Access .....	9
Searches.....	10
Viewer Integration.....	11
<b>DocuPhase SOAP Services API</b> .....	<b>13</b>
DocuPhase Application Server.....	13
DocuPhase DocumentServer .....	13
Working with DocuPhase SOAP Services .....	14
<b>DocuPhase HTML5 Viewer JavaScript API</b> .....	<b>16</b>
Access the JavaScript Client Integration Files .....	16
Requirements for HTML5 Viewer Integration .....	17
Steps to Integrate the HTML5 Viewer .....	17
<b>DocuPhase REST Services API</b> .....	<b>18</b>
Working with DocuPhase REST API Services .....	18
<b>Appendices</b> .....	<b>20</b>
Appendix A – Backward Compatibility & IIS Configuration .....	20
Appendix B – Examples of Invalid Command Line Parameters.....	21

## Revision History

Item #	Release #	Revision Date	Distrib/Public Date	Description	Tracking Notes
000	6.1 & 6.2	11/17/17	11/17/17	ORIGINAL	US 24845
001		12/8/17	02/09/18	Added information for URI format exception for Viewer to page 11 – , but not distributed in anticipation of additional changes coming for the Viewer section (see next item).	Item 26069
002		02/09/18	02/09/18	<ul style="list-style-type: none"><li>• Added Viewer Integration section (beginning on page 11)</li><li>• Added Security Services for REST services.</li></ul>	<ul style="list-style-type: none"><li>• Item 26724</li><li>• Requested by Felix during scrum on 02/08/18</li></ul>
003		02/13/18	02/13/18	<ul style="list-style-type: none"><li>• Corrected URLs from “DocuPhase38” to “DocuPhase” (on page 9)</li></ul>	As per request of S. Lincoln

## PURPOSE OF THIS GUIDE

The intended audience for this guide is software development professionals. Beginning with less technical information (such as interactions with DocuPhase via command line usage of products), this guide becomes progressively more technical. Software development professionals who review this document will then have necessary information to allow for the implementation of customized, integrated solutions.

An **SDK** is a set of software development tools that allows for the creation of applications in association with a specific software package. This means that software developers can create and implement customized functionality with DocuPhase: meeting specific business needs.

An **API** is an application programming interface, that allows software developers to write code that consumes the API. By consuming the API, developers can integrate software systems together without having to know or understand the inner workings of the software system being consumed.

## DOCUPHASE SDK OVERVIEW

The DocuPhase SDK provides a variety of technology options allowing interaction or integration between DocuPhase and other software systems.

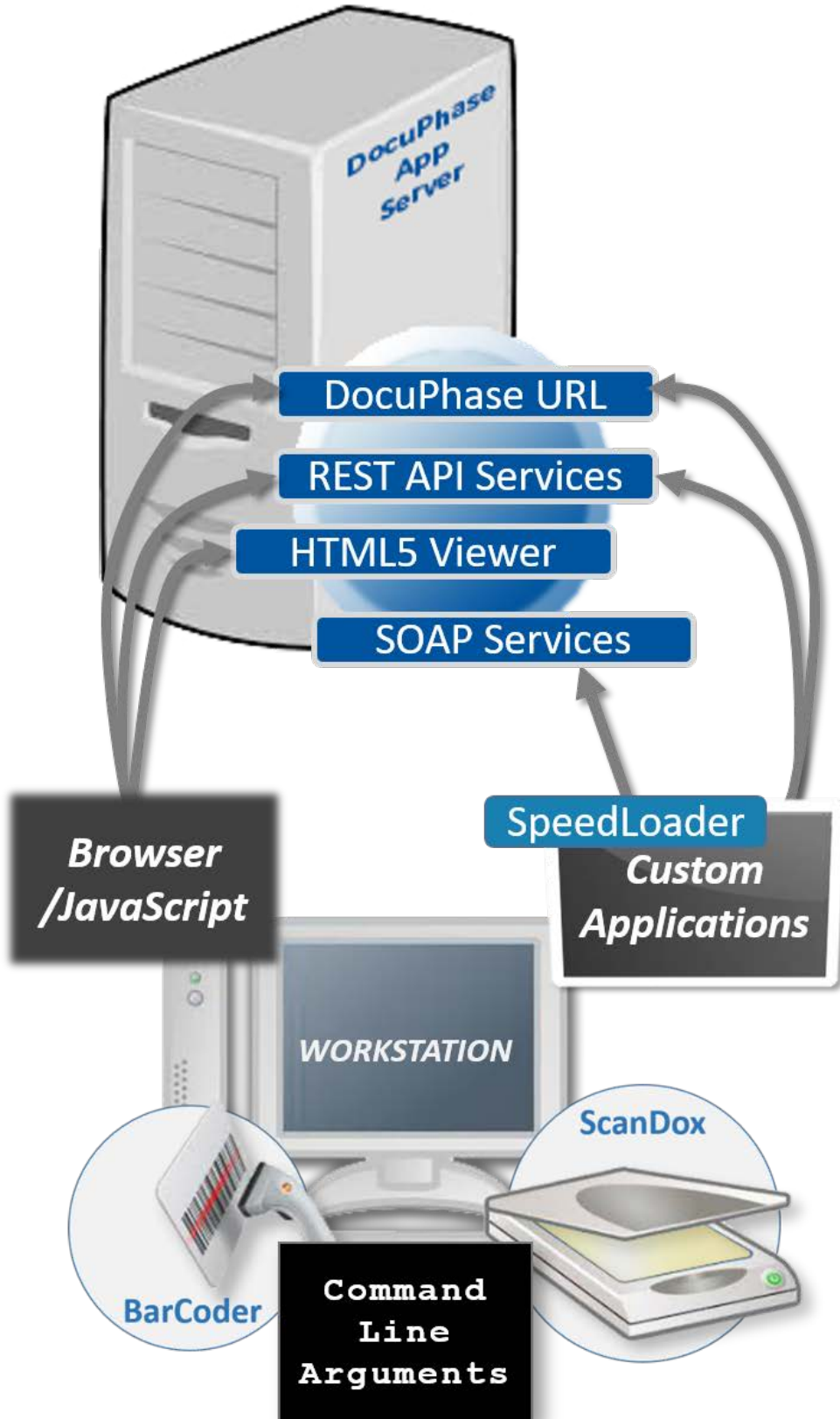
**Using the SDK, you can easily build applications that provide the following capabilities:**

- Document management, search, retrieval, and viewing
- Work item search
- Scanning
- Barcode processing
- Document and metadata import/export
- Business process management

**The DocuPhase SDK supports the following technologies:**

- API Services – Representational State Transfer API that provides access to DocuPhase logic and data

## SDK/API System Overview



# APPLICATION COMMAND LINE ARGUMENTS

## Basic Command Line Arguments Used for Both ScanDox and Barcoder

There are seven basic command argument elements used for Barcoder and ScanDox:

Argument Elements	Definition
<b>-a</b>	Load DocuPhase Application
<b>-d</b>	DocuPhase Web Server System Alias <ul style="list-style-type: none"> <li>This name appears within the Server name that appears in the drop-down list on the DocuPhase login screen.</li> <li>This element crosses references and uses the associated web server in the application.</li> </ul>
<b>-f</b>	Load file in DocuPhase
<b>-i</b>	DocuPhase Index Names and Values
<b>-p</b>	DocuPhase User Password
<b>-u</b>	DocuPhase User
<b>-w</b>	Attach DocuPhase Web Server

- If the User Name (-u), Password (-p), WebServer (-w) or Application ID (-a) parameters are missing on the command line, a message displays an error, and command line parameters are not processed.
- If any of the command line parameters are invalid, a message appears describing the error; and command line parameters are not processed.

## Working with Login Credential Line Arguments

Proper credentials are required for any operation in any application. In fact, logging into either Barcoder or ScanDox is automatic when the associated user and server credentials are entered. However, while automatic login is not unusual, there are other forms of command line options that can be used:

If Single Sign-On Mode is...	If Alias Server is...	Use this Command Line Argument...
OFF	DEFINED	<b>-u, -p and -w</b>
OFF	UNDEFINED	<b>-u, -p, -w, and -d</b>
ON	DEFINED	<b>-u, -w, and -d</b>
ON	UNDEFINED	<b>-u and -w</b>

### EXAMPLES

Here, the Command Line Argument displays automatic submission of credentials when not in Single Sign-On Mode using the web server name:

```

Microsoft Windows [Version 10.0.16299.19]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Documents and Settings\Administrator>cd\
C:\>cd\program files\docuphase\scan
C:\>DocuPhase\Scan>escan -usysadmin -pespeed -wdocuphasemdevprg1
C:\>DocuPhase\Scan>

Microsoft Windows [Version 10.0.16299.19]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Documents and Settings\Administrator>cd\program files\docuphase\scan
C:\>DocuPhase\Scan>escan -usysadmin -pespeed -wdocuphasemdevprg1 -ddev34
C:\>DocuPhase\Scan>
  
```

```

Command Prompt
Microsoft Windows [Version 10.0.16299.19]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Documents and Settings\Administrator>cd\
C:\>cd\program files\docuphase\scan
C:\>DocuPhase\Scan>escan -usysadmin -pespeed -wdocuphasemdevprg1
C:\>DocuPhase\Scan>
  
```

While here, the Command Line Argument displays an automatic submission of credentials when in Single Sign-On Mode using an alias.

Command Line Arguments for both Barcoder and ScanDox (continued on next page)

Command Line Arguments for Both Barcoder and ScanDox (continued)

### Creating/Editing Application, Indexes, and Index Value Line Arguments

Besides using the basic Command Line Arguments (shown on the previous page), settings can be created that automatically load the DocuPhase application, as well as its index values:



**IMPORTANT!**

- The Command Line Index Arguments use the Application ID, rather than the DocuPhase names. In other words, if you want to load an application named in DocuPhase as "Invoice", do not use the name "Invoice"; instead, use the Application ID found at Admin ► Application.

Name	Description	Implemented	Rev. Control	Rev. Tracking	Doc. Level Sec.	Workflow	Incoming Status	ID
GeneralCapture	General Capture Bin Application						E	
Invoice	All vendor invoices						E	3
Payroll	For screen captures and such						E	2

Annotations: A box labeled "NO" points to the "Name" column header. A box labeled "YES" points to the "ID" column header. The "Invoice" row is highlighted.

- You need to add the "-i" command for each index you want to add and define from the associated applications. Multiple index values can be included in a Command Line Argument.



**EXAMPLE**

This example provides you with an idea of how the indexes must be entered.

```
Microsoft Windows [Version 10.0.16299.19]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Documents and Settings\Administrator>cd\program files\docuphase\scan
C:\>cd\program file\docuphase\scan
C:\>DocuPhase\Scan>escan -iidx_name="john"
C:\>DocuPhase\Scan>
```

Annotations: A box labeled "Application" points to the path "C:\>cd\program file\docuphase\scan". A box labeled "Index" points to the parameter "-iidx\_name". A box labeled "Include Value" points to the value "john".

## Working with ScanDox

### Setting Up ScanDox-specific Line Arguments (File Names and Profiles)

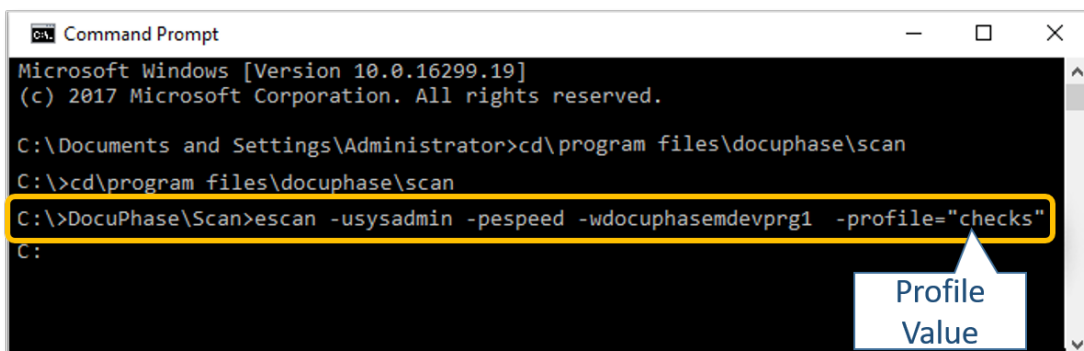
In addition to the Basic Command Line Arguments (as shown on page 3 of this guide), you can use others argument elements to automatically load file names and profiles for ScanDox.

Argument Elements	Definition
<b>-f</b>	Load file name
<b>-profile</b>	Load ScanDox profile (please refer to ScanDox user documentation for more information about profiles)



#### EXAMPLE

*This example provides you with an idea of how the indexes must be entered.*



```
Command Prompt
Microsoft Windows [Version 10.0.16299.19]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Documents and Settings\Administrator>cd\program files\docuphase\scan
C:\>cd\program files\docuphase\scan
C:\>DocuPhase\Scan>escan -usysadmin -pespeed -wdocuphasemdevprg1 -profile="checks"
C:
```

Profile Value



## Working with BarCoder

### Process Behaviors

- Only one BarCoder process runs at a time. Subsequent BarCoder processes pass the command line parameter values to the existing BarCoder process, and then terminate.
- Any data displayed on the existing BarCoder screen is replaced by command line parameter values that are passed. If a subsequent BarCoder process has no command line parameters, it terminates: having no effect on the existing BarCoder process currently running.
- The Command Line Processing check box appears at the top of the screen when an existing BarCoder process receives command line parameters.
- When Command Line processing is in effect, the application selection drop-down list is disabled (see *Selecting Another Application* shown below).
- When BarCoder receives field values from the Command Line parameter (-i), it displays any values passed, and the cursor positions itself on the first field on the screen.
- Within the **Tools>Display Fields** settings:
  - Fields set to display by default display regardless of whether or not the field is passed from the Command Line parameters.
  - Fields set to **not** display by default are only displayed if passed from the Command Line parameters.

### Setting New Default Values

In general, settings are not saved as new default values for subsequent barcode pages. To set new default values, follow the procedure below:

- 1) Deselect the Command Line processing option to disable Command Line processing.
- 2) Make changes to Print/Page settings or Tools settings.
  - *The BarCoder processes normally, and saves changes made as new defaults for subsequent barcode pages.*

### Selecting Another Application

- 1) Deselect the Command Line processing option to disable Command Line processing, remove the checkbox, then enable the drop-down list.

*To use a database lookup for a passed lookup field value:*

- 2) Position the cursor over the lookup field, then press the TAB key to initiate a database lookup and override values passed from the Command Line.

**-OR-**

*To make other field value changes (without performing a database lookup):*

Click in the field you want to change (DO NOT PRESS TAB), then make the change(s).

- *Repeat this step for all fields you want to edit.*

**-OR-**

*To print/preview the barcode page (without performing a database lookup):*

Click the **Print** or **Preview** options (DO NOT PRESS TAB).



#### **IMPORTANT!**

*When Command Line processing is in effect, any changes to Print/Page settings or changes made under the Tools menu item are only in effect for the current barcode page to be printed.*



#### **TIPS**

- *To keep the data passed, select a field or the **Print/Preview** option.*
- *Press the TAB key while in a lookup field to override any data passed with database values.*

# DOCUPHASE SPEEDLOADER

## Working with SpeedLoader

SpeedLoader is a .NET assembly (i.e., partially compiled code used for deployment) to import large numbers of documents into DocuPhase, in a transactional manner (i.e., at least partially in a sequential order to maintain consistency). In addition, SpeedLoader uses a specific import file as the source for documents to be imported: allowing for rollbacks as needed (see *NOTE* below).



### NOTE

Using an import file as a source allows for rollbacks of documents included in a failed import process. Imports using SpeedLoader are all or nothing; if any document in an import fails, the entire import is rolled back.

## Setting Up the Import File

The following are all characteristics of the SpeedLoader Import file (as described above):

- An XML file that defines the import and each document
- The first parameter is an import filename
- It contains two sections:
  - **Configuration** - appears once per file and contains configuration information. It also contains the following required elements:

Elements	Definition
<b>Server</b>	<ul style="list-style-type: none"> <li>• The name of the DocuPhase authentication server.</li> <li>• The value can be an FQDN (Fully Qualified Server Name) or an ordinary server name.</li> <li>• If the value is NOT an FQDN, then the assumed authentication is as follows: <a href="http://{server name}/applicationserver/authenticationservice.asmx">http://{server name}/applicationserver/authenticationservice.asmx</a></li> </ul>
<b>Table_name</b>	The name of the table into which a document is being placed
<b>Status</b>	The status assigned to the newly imported documents (e.g., XX, XX, XX)
<b>Username</b>	The name of the user to which the newly imported documents are assigned.

- **Document** - appears once per file, and can contain multiple document elements that describe the document being imported. Each document element is made up of one or more field elements that contain the following required attributes:

Elements	Definition
<b>Name</b>	The name of the column in the DocuPhase table into which the document is being placed
<b>Type</b>	The type of column (i.e., float, int, string, and date) into which a document is being placed
<b>Value</b>	<ul style="list-style-type: none"> <li>• The value being placed in the above named column.</li> <li>• All values must be valid for conversion to the specified type (as described above)</li> <li>• Dates must be in the format "YYYYMMDD"</li> </ul>
<b>itemcount</b>	Informs the system of the number of document elements are designated
<b>doc_file_name</b>	This points to the file belonging to the document element.



### EXAMPLE

```
<data version="1.0.0" type="iDatix.SpeedLoader">
  <configuration>
    <server>localhost</server>
    <table_name>obj_2</table_name>
    <status>E</status>
    <username>sysadmin</username>
  </configuration>
  <documents>
    <document>
      <fields>
        <field name="idx_name" type="string" value="john" />
        <field name="idx_amount" type="float" value="123.45" />
        <field name="idx_count" type="int" value="100" />
        <field name="idx_date" type="date" value="20040609" />
      </fields>
      <doc_file_name>c:\dir\filename00001.tif</doc_file_name>
    </document>
  </documents>
  <itemcount>1</itemcount>
</data>
```

Working with SpeedLoader (continued on next page)

### Working with SpeedLoader (continued)

## Making Pre-Import Confirmations

Before SpeedLoader attempts to import documents, it automatically performs precise processes to help prevent any import from failing. These processes include the following:

- Each import file is checked to verify that all index values can be converted to the designated type. If any value fails the conversion, SpeedLoader doesn't attempt the import; validation includes checking string field length.
- the total files size for all documents is calculated, then doubled, and submitted for import to make sure that there is enough space to complete the import process. If SpeedLoader determines that there isn't enough space for the import of all the documents, the import is not performed.

## Using SpeedLoader Interface Command Elements

The Interface Command is a combination of the following elements:

Elements	Definition
<b>FileName</b>	Always the first parameter of the command, and is a required element
<b>Username</b>	The DocuPhase user to impersonate
<b>Password</b>	The password associated with the user (as described above)
<b>Instance</b>	An optional value that corresponds to the instance names in the DocuPhase authentication server. If no Instance value is provided, the "Default" instance is used.



### EXAMPLE

Here's how the Interface Command Elements appear:

```
SpeedLoader.Process(string filename, string username, string password, string instance)
```

## SETTING UP AND USING THE DOCUPHASE URL API

This section describes how the system provides third-party user (i.e., someone outside your designated user group) access to one or more documents from within the DocuPhase platform. In this process, the DocuPhase URL automatically logs the user into the system, allows him/her to search for (and select) a document that is then retrieved from the DocuPhase server. The document is then rendered and displayed in the DocuPhase Viewer (also see *NOTES* below).

- System access to users (i.e., logging in) – see below
- Searches – see page 10
- Viewer Integration – see page 11



### NOTES

- *DocuPhase can interface with any ODBC compliant database for data exchange and third-party product integration.*
- *The Document Viewer can be embedded into a third party application: giving users DocuPhase Viewer Toolbar functionality.*
- *See Appendix A on page 20 of this document for information about Backward Compatibility.*

## System Access

The login prompt appears when a user is not already logged into DocuPhase. In order for a third-party user to bypass this page, and continue through to the requested search, the user must include the key/values associated with the Access.aspx page within the QueryString (see details below).

### Access.aspx

Querystrings must include the following:

- **Action** = Login (required)
- **UserName** = USERNAME (required)
- **Password** = PASSWORD (required) OR EncryptedPassword = ENCRYPTEDPASSWORD



### EXAMPLE

- `http://SERVERNAME/DocuPhase/web/authorization/Access.aspx?Action=login&UserName=USERNAME&Password=PASSWORD`
- `http://SERVERNAME/DocuPhase/web/authorization/Access.aspx?Action=login&UserName=USERNAME&EncryptedPassword=ENCRYPTEDPASSWORD`

## HandlerPage.aspx

The HandlerPage is the component that processes all queries from outside of DocuPhase. This is the page that is accessed using appropriate parameters; including parameters for logging in, as well as for searching (see **Searches** section on page 10 for more about each search, and its required parameters).



### EXAMPLE

- `http://SERVERNAME/DocuPhase/web/handler/HandlerPage.aspx?Action=login&UserName=USERNAME&PASSWORD=PASSWORD&ApplicationID=2`
- `http://SERVERNAME/DocuPhase/web/handler/HandlerPage.aspx?Action=login&UserName=USERNAME&Password=PASSWORD&ApplicationID=2&IDX_Name=fredex`

## Searches

Third-party users can utilize a set of QueryString values that perform searches - just as if the user was searching from inside DocuPhase. There are three types of searches a user can perform:

- **“CrossQuery” Search** – performs a search across all applications based on a specific index value entered in the QueryString
- **Global Search** – performs a search of every document, in every application, for specific keyword values entered in the QueryString
- **Generic Search** – performs a search using a combination of Application, ObjectID, index value, and/or index value filter according to the values entered in the QueryString

**IMPORTANT!**  
 Please remember... Entries for Search and Indexes ARE CASE SENSITIVE!

## Search QueryString Values

Search Type	Querystring Value*	Required Values
CrossQuery	<b>index, search</b>	Values shown here in <b>red</b> are required.
Global	<b>gsearch</b>	
Generic	<b>ApplicationID, IDX_IndexName, ObjectID</b>	

## Search Hierarchy

Searches are processed in the following order:

- 1<sup>st</sup> - CrossQuery search
- 2<sup>nd</sup> - Global search
- 3<sup>rd</sup> - Generic search

**EXAMPLE**  
 IF the following query was entered:  
 http://SERVERNAME/DocuPhase/web/handler/HandlerPage.aspx?Action=login&USERNAME=USERN  
 AME&PASSWORD=PASSWORD&ApplicationID=2&gsearch=Apple&search=Orange&index=IDX\_Fruit  
**THEN** the cross query result will be recognized and executed, whereas the global search will not be  
 recognized or executed.

## In/Not in Searches

Within the Search querystrings, you can specify the inclusion or exclusion of certain items included in the search results:

To...	Use this Querystring Value
<b>Include results</b>	IDX_INDEXNAME=in(value1,value2,...,valueN)
<b>Exclude results</b>	IDX_INDEXNAME=not in(value1,value2,...,valueN)

**EXAMPLE**

- To include documents from the White Orchid and Fredex applications, with a status of “paid” enter the following:  
 http://SERVERNAME/DocuPhase/web/handler/HandlerPage.aspx?Action=login&UserName=USERNAME&Passw  
 ord=PASSWORD&ApplicationID=17&IDX\_name=in(white orchid,fredex)&idx\_status=in(paid)
- To include documents from any applications except those associated with “Dayback” and “Fredex” enter the  
 following: http://SERVERNAME/DocuPhase/web/handler/HandlerPage.aspx?Action=login&UserName  
 =USERNAME&Password=PASSWORD&ApplicationID=2&IDX\_Name=not in(dayback,fredex)
- To include documents from the Fredex application only, enter the following:  
 http://SERVERNAME/DocuPhase/web/handler/HandlerPage.aspx?Action=login&UserName  
 =USERNAME&Password=PASSWORD&ApplicationID=2&IDX\_Name=in(fredex)

*Searches (continued on next page)*

*Searches (continued)***Required Licensing**

To maintain the integrity of the DocuPhase system, as well as to safeguard client information, the SDK URL-based searching requires a license. The product under which this URL search is licensed is “Partner Extension”, and includes the following properties:

Search Type	Status	Property
<b>CrossQuery</b>	Pending Approval	CrossQuerySearchEnabled
<b>Global</b>	Pending Approval	GlobalSearchEnabled
<b>Generic</b>	Current standard search	GenericSearchEnabled

*NOTES*

- Names of Search Properties may need to change to obscure identity, if required.
- Each property maintains a Boolean value, where:
  - **'True'** means it is enabled.
  - **'False'** means it is disabled.

**Viewer Integration****URL Formats**

You can use query string values to load the ActiveX Document Viewer using the following URL formats:

For an application and document with Revision Control...	Use this URL format...
<b>Enabled</b> ( <input checked="" type="checkbox"/> )	<code>https://server.mydomain.com/docuphase/web/ /authorization/Access.aspx?ReturnUrl=/iSynergy/web/search/viewer.aspx?appId%3d2%26objectId%3d47%26version%3d0</code>
<b>Disabled</b> ( <input type="checkbox"/> )	<code>https://server.mydomain.com/docuphase/web/ /authorization/Access.aspx?ReturnUrl=/iSynergy/web/search/viewer.aspx?appId%3d2%26objectId%3d47</code>

## Viewer Integration (continued)

### Query String Parameters

The query string parameters are as follows:

This value...	Refers to the ...
<b>ReturnUrl</b>	Encoded DocuPhase URL that will be loaded (see <i>IMPORTANT!</i> note below)
<b>appId</b>	application ID that contains the document.
<b>objectId</b>	object ID (document ID) for the document to be viewed.
<b>version</b>	version number of the document if the application is revision controlled.



#### *IMPORTANT!*

The *Return Url* parameter (as shown above) is encoded to prevent potentially malicious data from being passed to DocuPhase.

The following example demonstrates how the DocuPhase ActiveX Viewer can be loaded from an external software system: producing a seamless integration with DocuPhase.

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title></title>
</head>
<body>

  <script>
    var newWindow;

    // works
    //newWindow =
window.open('https://myServer/iSynergy/web/authorization/Access.aspx?ReturnUrl=/iSynergy/web/search/viewer.aspx?appId%3d2%26objectId%3d47', '_blank');

    // works
    var newURL =
'https://myServer/iSynergy/web/authorization/Access.aspx?ReturnUrl=/iSynergy/web/search/viewer.aspx?' + encodeURIComponent('appId=2&objectId=47');

    alert(newURL);

    newWindow = window.open(newURL, '_blank');

    newWindow.focus();

  </script>

</body>
</html>
```

## DOCUPHASE SOAP SERVICES API

DocuPhase includes a number of “Servers” that provide one or more SOAP (i.e., Simple Object Access Protocol) services. By utilizing the DocuPhase SOAP services API integrated systems execute logic built in to DocuPhase: ensuring data integrity.

**NOTE**

*DocuPhase SOAP Services API is considered a legacy technology, and is, therefore, gradually being deprecated.*

All SOAP services are discoverable and publish WSDL documents. To access the WSDL document for a particular service, use the following URL pattern to get to the WSDL of the corresponding service:

<http://localhost/{servername}/{servicename}.asmx?WSDL>

### DocuPhase Application Server

This section provides a description of each “Server” and its corresponding services. DocuPhase Application Server contains two SOAP services that provide access to application data:

#### DataService

The DataService SOAP service provides access to retrieve data, as well as some additional service methods that allow for update of system level information.

To access the DataService WSDL\* that describes each service method along with required parameters, use the following URL format:

<http://{machine name}/ApplicationServer/DataService.asmx?WSDL>

#### AuthenticationService

The AuthenticationService SOAP service provides methods for authentication and authorization.

To access the AuthenticationService WSDL\* that describes each service method along with required parameters, use the following URL format:

<http://{machine name}/ApplicationServer/AuthenticationService.asmx?WSDL>

### DocuPhase DocumentServer

The DocuPhase Document Server contains a single SOAP service called DocumentService that provides access to documents.

#### Using DocumentService

The DocumentService SOAP service provides secure access to documents in the DocuPhase repository. This service leverages SOAP service standards such as ws-security, ws-policy and ws-attachments.

To access the DocumentService WSDL\* that describes each service method along with required parameters, use the following URL format:

<http://{machine name}/DocumentServer/DocumentService.asmx?WSDL>

*\*Please refer to the service WSDL to view details of each method, the input parameters, and a description of what action the method performs.*

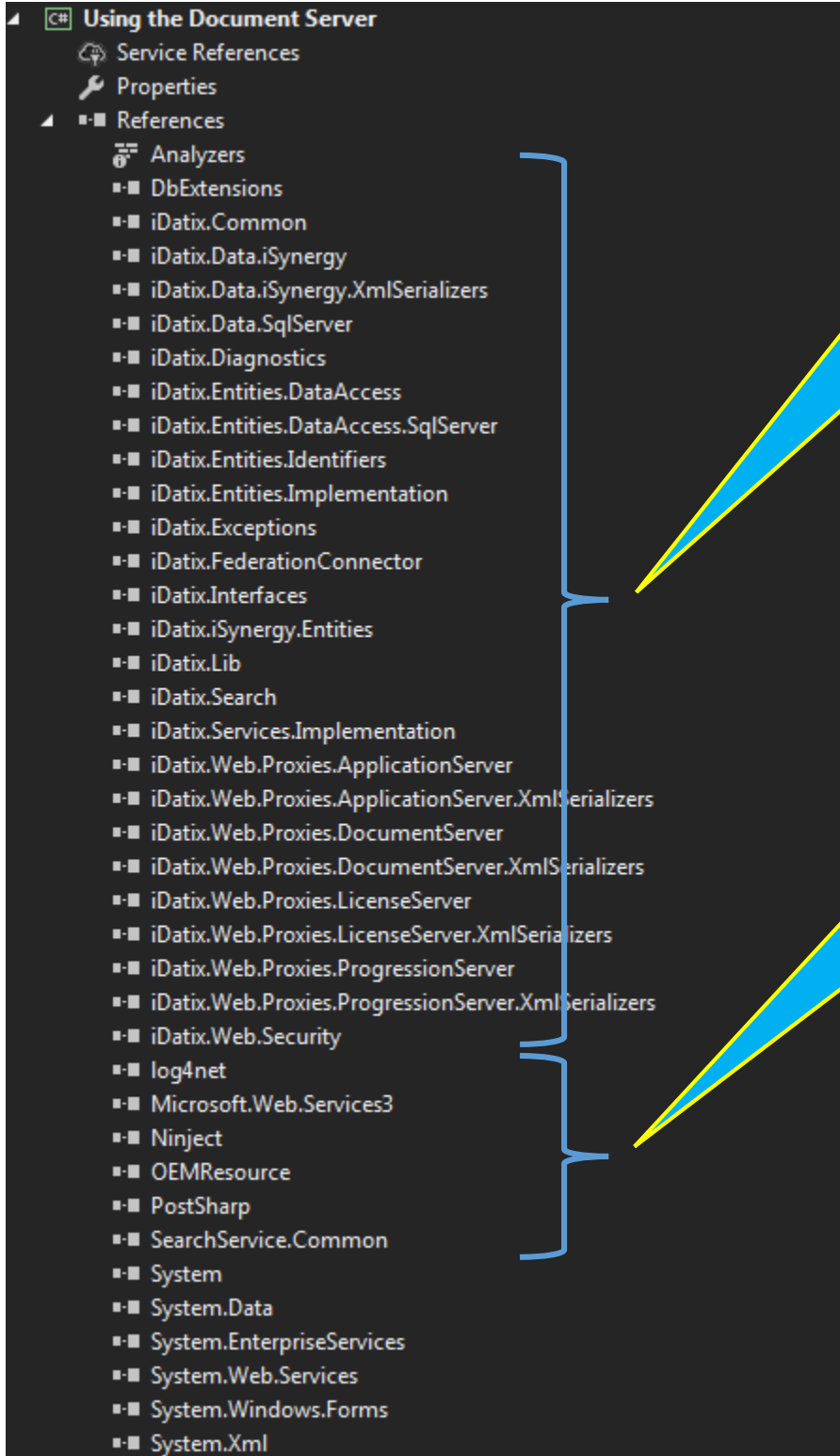


## Working with DocuPhase SOAP Services

The DocuPhase SDK provides sample code that demonstrates the use of the DocumentServer DocumentService. This sample code is made available as part of an installable MSI package, and is provided upon request.

### Project References

The project references DocuPhase assemblies required in order for the sample code to compile and execute. These references are shown below:



Required DocuPhase assemblies begin with iDatix. naming convention.  
DbExtensions is a DocuPhase assembly.

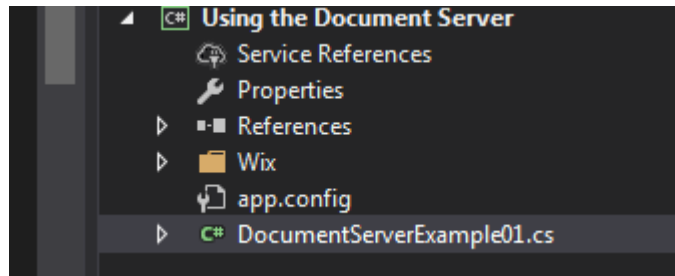
Additional required assemblies log4net, Ninject, PostSharp.  
SearchService.Common is a DocuPhase assembly.  
OEMResource is a DocuPhase assembly.

Working with DocuPhase SOAP Services (continued)

## Working with DocuPhase SOAP Services (continued)

### Sample Code Explanation

The sample code located in a project called *Using the Document Server* found in a single class file `DocumentServerExample01.cs` (as shown below):



The source code within this file demonstrates how to utilize the ApplicationServer DataService as well as the DocumentServer DocumentService, and performs the following actions:

- ✓ Queries the DataService
- ✓ Checks out a document from the DocumentService
- ✓ Checks in a document to the DocumentService

In order to perform these actions, the code performs the following detailed steps:

- 1) Creates proxy instances to provide connectivity to the DocumentService and DataService.
- 2) Sets URLs to the services for the proxies to use.
- 3) Authenticates a DocuPhase user against the AuthenticationService, and retrieves an authentication token.
- 4) Sets up the DocInfo object with valid data.
- 5) Calls the services with the DocInfo object to manipulate data within the DocuPhase repository.



#### NOTE

*If you want to build your own project that interacts with the DocuPhase repository via .NET code, this project is a great starting point that should provide functionality that most systems need.*

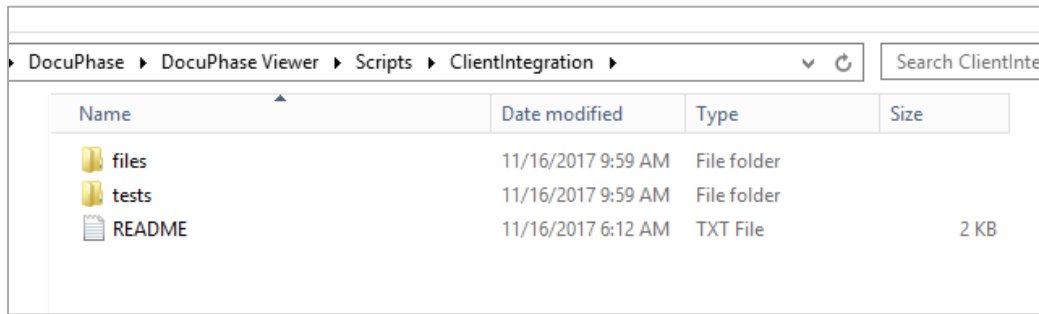
## DOCUPHASE HTML5 VIEWER JAVASCRIPT API

DocuPhase provides an HTML5 Document Viewer capable of integration with your own web applications. Sample code and a Readme file are installed with the product on the DocuPhase web server.

### Access the JavaScript Client Integration Files

*On your server:*

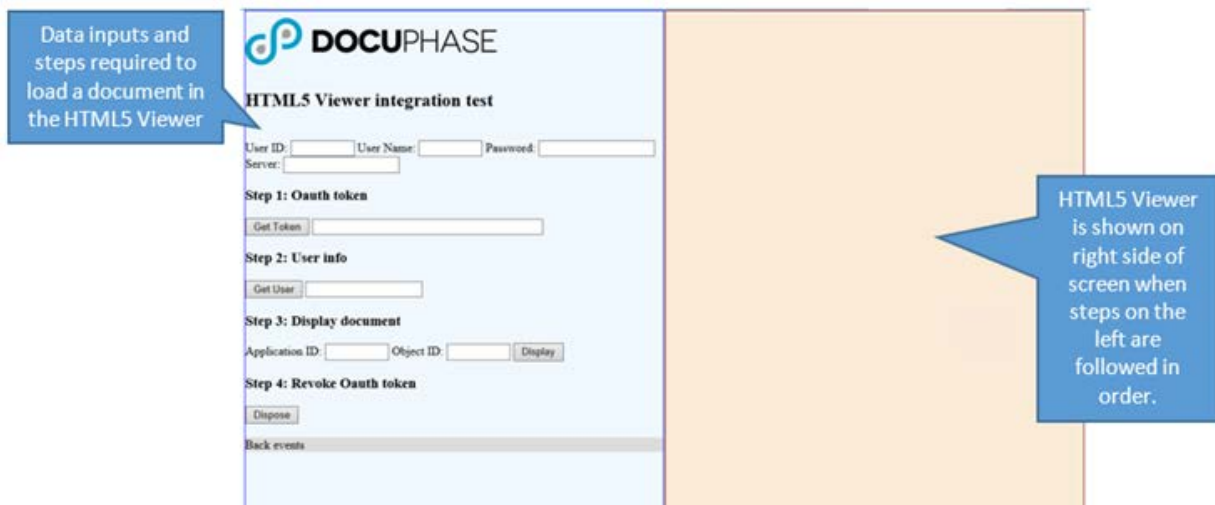
- 1) Navigate to the location at which DocuPhase is installed (this example assumes C:\Program Files\DocuPhase).
- 2) Go to the DocuPhase Viewer\Scripts\ClientIntegration directory:



- The README file contains notes about how to integrate the HTML5 Viewer into your application.
- Within the tests folder, there are two files:
  - DocuPhase.HTML5.Viewer.TestScript.js
  - DocuPhase.HTML5.Viewer.TestPage.html
- The test page has the HTML markup and JavaScript file references needed in order to allow the page to interact with the DocuPhase REST API and the HTML 5 Viewer:

```
<head>
<title></title>
<script type="text/javascript" src="../../files/jquery-1.11.0.min.js"></script>
<script type="text/javascript" src="../../files/pubsub.min.js"></script>
<script type="text/javascript" src="../../files/DocuPhase.HTML5.Viewer.Client.js"></script>
<script type="text/javascript" src="../../files/DocuPhase.HTML5.Viewer.ClientBack.js"></script>
<script type="text/javascript" src="../../files/DocuPhase.HTML5.Viewer.ClientSettings.js"></script>
<script type="text/javascript" src="DocuPhase.HTML5.Viewer.TestScript.js"></script>
<meta charset="utf-8" />
</head>
```

*The resulting page is rendered as such:*



**Data inputs and steps required to load a document in the HTML5 Viewer**

**DOCUPHASE**

**HTML5 Viewer integration test**

User ID:  User Name:  Password:   
 Server:

**Step 1: Oauth token**

**Step 2: User info**

**Step 3: Display document**

Application ID:  Object ID:

**Step 4: Revoke Oauth token**

Back events

**HTML5 Viewer is shown on right side of screen when steps on the left are followed in order.**

## Requirements for HTML5 Viewer Integration

- ✓ The hosting Web Application and the DocuPhaseHTML5Viewer must be installed in the same domain.
- ✓ The DocuPhase DataService REST service must be installed.
- ✓ The DocuPhase Security REST service must be installed.

## Steps to Integrate the HTML5 Viewer

- 1) The JavaScript files listed below must be included in the hosting Web Application.
- 2) Update DocuPhase.HTML5.Viewer.ClientSettings.JS to reflect your mechanism to set request information.
- 3) Update DocuPhase.HTML5.Viewer.ClientBack.js to reflect your mechanism to respond to each Viewer function completion.
- 4) Update DocuPhase.HTML5.Viewer.Client.js to reflect the HTML container that will be hosting the HTML5Viewer in the host application page.
- 5) Insert an HTML container (usually a DIV) in the host application page as specified in step 4.
- 6) Make sure DocuPhase.HTML5.Viewer.ClientSettings.JS is called to populate the \$.HTML5ViewerSettings object.
- 7) Call the corresponding Viewer function via the \$.HTML5Viewer method passing the needed parameters (see each method description in DocuPhase.HTML5.Viewer.Client.js).

File	Description
/Scripts/ClientIntegration/files/pubsub.min.js	Client side bus implementation
/Scripts/ClientIntegration/files/jquery-1.11.0.min.js	Client side JavaScript Framework
/Scripts/ClientIntegration/files/DocuPhase.HTML5.Viewer.Client.js	Client side HTML5 Viewer JavaScript API file. For more details check the comments in the file.
/Scripts/ClientIntegration/files/DocuPhase.HTML5.Viewer.ClientBack.js	Provides the mechanism to hook and respond to the HTML5 Viewer functions completion. Function completion available: CheckInComplete, CheckOutComplete, UndoCheckOutComplete, DeleteRevisionComplete, PromoteToTipComplete, Refresh. For more details check the comments in the file.
/Scripts/ClientIntegration/files/DocuPhase.HTML5.Viewer.ClientSettings.js	Provides the mechanism to set the request information. These include: Token (obtained from the Security Service), EmailPass flag (if user has e-mail password set in the DocuPhase system), Data Service instance number, Viewer URL. For more details check the comments in the file.

## DOCUPHASE REST SERVICES API

DocuPhase provides RESTful API services. By utilizing the DocuPhase REST services API, integrated systems are executing logic built in to DocuPhase: ensuring data integrity. DocuPhase REST Services API is considered current technology, and continues to be enhanced.

All REST API endpoints include documentation that may be accessed by using the following URL pattern:

<http://{server name}/{service name}/documentation>

The following table shows all services and their corresponding documentation URLs:

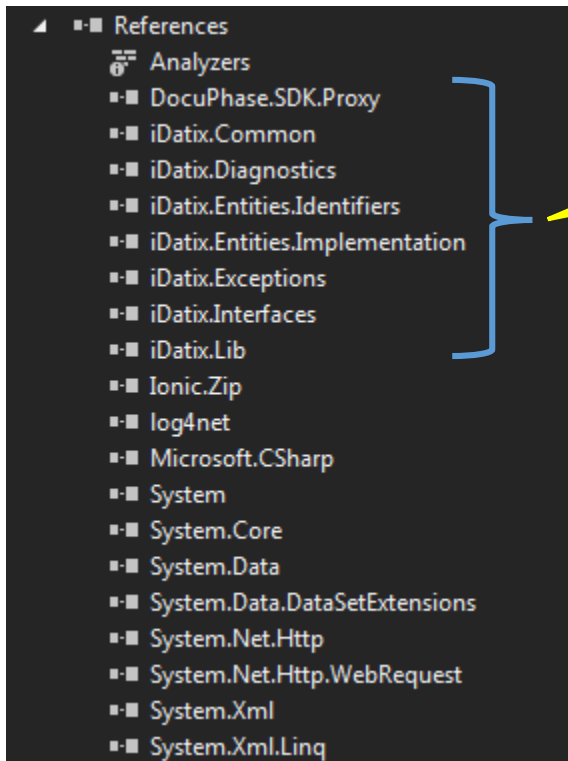
Service	Description	Documentation URL
ConfigurationService	Support for access to Configuration, Indexing and Product settings	<a href="http://{servername}/configuration-service/documentation">http://{servername}/configuration-service/documentation</a>
DataService	Support for access to information related to Documents (content, upload, versions), Forms, Annotations, OCR Data and Applications	<a href="http://{servername}/data-service/documentation">http://{servername}/data-service/documentation</a>
SearchService	Support for access to search for Documents and Workflow work items	<a href="http://{servername}/SearchService/documentation">http://{servername}/SearchService/documentation</a>
SecurityService	Support for security related activities, such as authentication, user/group maintenance and permissions	<a href="http://{servername}/SecurityService/documentation">http://{servername}/SecurityService/documentation</a>

## Working with DocuPhase REST API Services

The DocuPhase SDK provides sample code that is located in a project called *Using the RESful Web Services* that demonstrates use of the DocuPhase REST API Services from C# .NET. This sample code is provided upon request.

### Project References

The project references DocuPhase assemblies required in order to compile and execute as shown below:



Required DocuPhase assemblies begin with iDatix. naming convention.

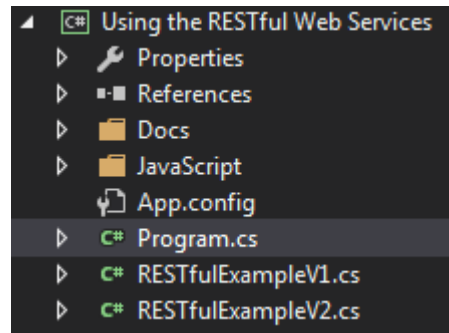
DocuPhase.SDK.Proxy is the DocuPhase proxy assembly which simplifies access to RESTful API services.

*Working with DocuPhase REST API Services (continued on next page)*

## Working with DocuPhase REST API Services (continued)

### Sample Code Explanation

The sample code located in a project called Using the RESTful Web Services project is found in three class files as shown below:



- Program.cs is the starting point for the application. This simple program instantiates classes found in RESTfulExampleV1.cs and RESTfulExampleV2.cs.
- RESTfulExampleV1 performs the following actions:
  - ✓ Logs in to DocuPhase using DocuPhase credentials
  - ✓ Gets applications and enumerates them
  - ✓ Creates a new document
  - ✓ Gets a document from the DataService
  - ✓ Writes the document to local disk
  - ✓ Checks out the document
  - ✓ Modifies some document content
  - ✓ Checks in the document
  - ✓ Demonstrates use of search to find documents in DocuPhase
- RESTfulExampleV2 performs the following actions:
  - ✓ Logs in to DocuPhase using DocuPhase credentials
  - ✓ Creates a document (non-revision controlled)
  - ✓ Updates document data



#### NOTE

- These code examples make use of the DocuPhase SDK Proxy which simplifies access to the REST API.
- Use of the SDK Proxy is not required in order to access the REST API.

## APPENDICES

### Appendix A – Backward Compatibility & IIS Configuration

#### Backward Compatibility

Although the API key/value pairs have changed, this API was designed to support API calls from prior DocuPhase versions. The table below lists the changes between 5.1 and prior versions.

Previous Versions	Current Version
<b>Current_App_ID</b>	ApplicationID
<b>User_ID</b>	UserName
<b>Logon</b>	Login
<b>EPass</b>	EncrypedPassword
<b>Pass</b>	Password

#### IIS Configuration

- All prior calls were made to Results\_Frame.asp. Those calls should now be set to request to HandlerPage.aspx with the appropriate QueryString values.
- Because the libraries controlling asp and asp.net applications are different, anything referencing the older Results\_Frame.asp page is not accessible from a 5.1 DocuPhase instance.
- To alleviate any problems for applications that still reference Results\_Frame.asp, the installer configures IIS to redirect all requests from Results\_Frame.asp to HandlerPage.aspx.

The following steps list the actions required to manually change IIS:

- 1) Go to Start ► Administrative Tools ► Internet Information Services IIS Manager.
- 2) Navigate to **Web Sites**.
- 3) Expand the **Default Web Site** folder, then select **DocuPhase** or **iSynergy**.
- 4) Locate, then right-click on **Results\_Frame.asp**.
- 5) Select **Properties**.

*In the Results\_Frame.asp properties dialog:*

- 6) Enable the **A Redirection to a URL** option.

*In the “Redirect to” field:*

- 7) Enter the following URL:  
<http://SERVERNAME/DocuPhase/web/handler/HandlerPage.aspx> (where SERVERNAME is the name of your server).
- 8) Click **Apply**.
- 9) Click **OK**, then close IIS Management Console.

## Appendix B – Examples of Invalid Command Line Parameters

- Invalid username, password or Web server.
- Application ID parameter missing, not found in DocuPhase database or Application referenced by the Application ID is not "implemented" in DocuPhase.
- Index field name/value parameter (-i) is not passed using the following format: -iIDX\_name=fieldvalue.
- Index field name does not exist within the Application ID.
- Index field value is invalid. The program tests for numeric, decimal and date/time values if applicable for the passed index field name.
- -iIDX\_TestDate="13/04/2008" is an invalid date.
- -iIDX\_TestDate="14/29/2008" is a valid date