



# Acrobat Search API Reference

Technical Note #5162

Version : Acrobat 6.0



**ADOBE SYSTEMS INCORPORATED**

**Corporate Headquarters**


345 Park Avenue

San Jose, CA 95110-2704

(408) 536-6000

<http://partners.adobe.com>

*May, 2003*



Copyright 2003 Adobe Systems Incorporated. All rights reserved.

NOTICE: All information contained herein is the property of Adobe Systems Incorporated. No part of this publication (whether in hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the Adobe Systems Incorporated.

PostScript is a registered trademark of Adobe Systems Incorporated. All instances of the name PostScript in the text are references to the PostScript language as defined by Adobe Systems Incorporated unless otherwise stated. The name PostScript also is used as a product trademark for Adobe Systems' implementation of the PostScript language interpreter.

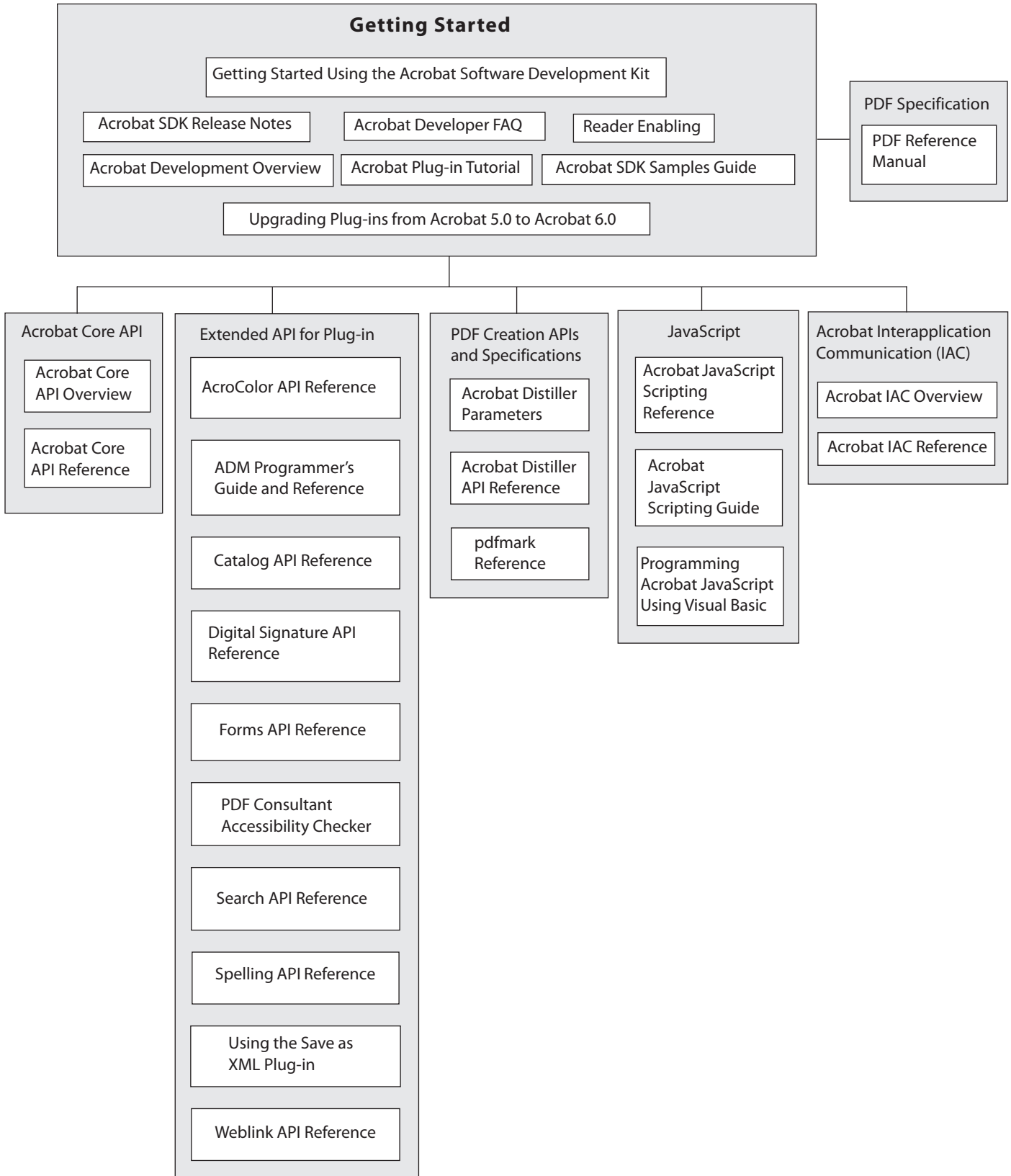
Except as otherwise stated, any reference to a "PostScript printing device," "PostScript display device," or similar item refers to a printing device, display device or item (respectively) that contains PostScript technology created or licensed by Adobe Systems Incorporated and not to devices or items that purport to be merely compatible with the PostScript language.

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Acrobat Capture, Distiller, PostScript, the PostScript logo and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apple, Macintosh, and Power Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. PowerPC is a registered trademark of IBM Corporation in the United States. ActiveX, Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. UNIX is a registered trademark of The Open Group. Verity is a trademark of Verity, Inc. Lextek is a trademark of Lextek International. All other trademarks are the property of their respective owners.

This publication and the information herein is furnished AS IS, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and noninfringement of third party rights.

# Acrobat SDK Documentation Roadmap







# Contents

<b>Preface</b> . . . . .	<b>7</b>
Contents . . . . .	7
Other Useful Documentation . . . . .	8
Conventions Used in This Book . . . . .	8
<b>Methods</b> . . . . .	<b>11</b>
SearchAddIndex . . . . .	11
SearchAddIndexEx . . . . .	12
SearchCountIndexList . . . . .	13
SearchExecuteQuery . . . . .	14
SearchExecuteQueryEx . . . . .	16
SearchGetIndexByPath . . . . .	19
SearchGetIndexByPathEx . . . . .	20
SearchGetIndexFlags . . . . .	21
SearchGetIndexList . . . . .	22
SearchGetIndexPath . . . . .	23
SearchGetIndexPathEx . . . . .	24
SearchGetIndexFileSys . . . . .	25
SearchGetIndexTitle . . . . .	26
SearchGetIndexTitleEx . . . . .	27
SearchGetNthIndex . . . . .	28
SearchIsLegacySearchAvailable . . . . .	29
SearchRemoveIndex . . . . .	30
SearchSetIndexFlags . . . . .	31
<b>Apple Events</b> . . . . .	<b>33</b>
SearchAddIndex . . . . .	33
SearchCountIndexList . . . . .	34
SearchDoQuery . . . . .	35
SearchGetIndexByPath . . . . .	37
SearchGetIndexFlags . . . . .	38
SearchGetIndexList . . . . .	39
SearchGetIndexPath . . . . .	40
SearchGetIndexTitle . . . . .	41
SearchGetNthIndex . . . . .	42
SearchRemoveIndex . . . . .	43
SearchSetIndexFlags . . . . .	44



**DDE Messages . . . . .45**

- Simple Query Item . . . . . 45
- Query Item . . . . . 45
- Manipulating Indices Through DDE . . . . . 48

**Lists . . . . .51**

- Menu Names. . . . . 51
- Menu Item Names . . . . . 51
- Toolbar Button Names . . . . . 52



# Preface

The Adobe® Acrobat® Search plug-in allows users to perform text searches in PDF documents. It adds menus, menu items, toolbar button, and a Search panel to the Acrobat viewer.

The Search plug-in exports a Host Function Table (HFT) containing several methods that can be used by other plug-ins.

The HFT's name is **Search**, and its version number is 0. To use the Search plug-in's HFT, a plug-in must include the header file **SrchClis.h**. The plug-in must also import the HFT using **ASExtensionMgrGetHFT** and assign the HFT returned by this call to a plug-in-defined global variable named **gSearchHFT**.

Acrobat 6.0 has two versions of the Search plug-in, Search (new in Acrobat 6.0) and Search5 (also available in Acrobat 5.0).

- The Search plug-in uses a search engine licensed from Lextek International. Lextek can be contacted at <http://www.lextek.com>.
- The Search5 plug-in uses a search engine licensed from Verity, Inc. Verity can be contacted at <http://www.verity.com>.

You can do the following with either version of the Search plug-in:

- Create or delete indexes
- Ask what indexes are available
- Send queries to an existing index

You cannot use either version of the Search plug-in to directly obtain the results of a search, for manipulation or for display in an application other than Adobe Acrobat.

Search supports interapplication communication (IAC) in the form of Apple events on the Macintosh and DDE messages under the Windows® operating system. These Apple events and DDE messages allow remote clients to submit search queries and manipulate a list of indexes (the list of indexes is referred to as the *shelf*).

This technical note describes the HFT and IAC APIs supported by the Acrobat Search plug-in. It also contains the names of items added to the Acrobat viewer user interface by the Search plug-in.

If you received this technical note without obtaining the entire Acrobat Software Development Kit (SDK), you can get the complete SDK by visiting:

<http://partners.adobe.com/asn/developer/acrosdk/main.html>

---

## Contents

This technical note contains the following sections:

- [Methods](#) describes in detail of each HFT method, including parameters and return value.
- [Apple Events](#) describes in detail each Apple event, including parameters and return value.
- [DDE Messages](#) describes in each DDE message and its arguments.
- [Lists](#) describes Search dialog boxes, menu item names, and toolbar button names.

---

## Other Useful Documentation

You should be familiar with the Acrobat core API and Interapplication Communication (IAC) API. The following technical notes provide this information.

[Acrobat Core API Overview](#), Technical Note #5190. Gives an overview of the objects and methods provided by the Acrobat Core API.

[Acrobat Core API Reference](#), Technical Note #5191. Describes in detail the objects and methods provided by the Acrobat core API.

[Acrobat Interapplication Communication Overview](#), Technical Note #5164. Contains an overview of the Apple event, OLE method, and DDE message interfaces provided to the Acrobat viewer.

[Acrobat Interapplication Communication Reference](#), Technical Note #5165. Contains detailed information on the Apple event, OLE method, and DDE message interfaces provided to the Acrobat viewer.

[PDF Reference](#), Third Edition, version 1.4. Provides a description of the PDF file format, as well as suggestions for producing efficient PDF files. It is intended for application developers who wish to produce PDF files directly.

---

## Conventions Used in This Book

The Acrobat documentation uses text styles according to the following conventions.

Font	Used for	Examples
monospaced	Paths and filenames	<code>C:\templates\mytmpl.fm</code>
	Code examples set off from plain text	These are variable declarations: <code>AVMenu commandMenu, helpMenu;</code>



Font	Used for	Examples
monospaced bold	Code items within plain text	The <code>GetExtensionID</code> method ...
	Parameter names and literal values in reference documents	The enumeration terminates if <code>proc</code> returns <code>false</code> .
monospaced italic	Pseudocode	<code>ACCB1 void ACCB2 ExeProc(void)</code> <code>{ do something }</code>
	Placeholders in code examples	<code>AFSimple_Calculate(cFunction,</code> <code>cFields)</code>
blue	Live links to Web pages	The Acrobat Solutions Network URL is: <a href="http://partners/adobe.com/asn/">http://partners/adobe.com/asn/</a>
	Live links to sections within this document	See <a href="#">Using the SDK</a> .
	Live links to other Acrobat SDK documents	See the <a href="#">Acrobat Core API Overview</a> .
	Live links to code items within this document	Test whether an <code>ASAtom</code> exists.
bold	PostScript language and PDF operators, keywords, dictionary key names	The <code>setpagedevice</code> operator
	User interface names	The <b>File</b> menu
italic	Document titles that are not live links	<i>Acrobat Core API Overview</i>
	New terms	<i>User space</i> specifies coordinates for...
	PostScript variables	<i>filename</i> <code>deletefile</code>



## Preface

*Conventions Used in This Book*

# Methods

---

## SearchAddIndex

```
SearchIndexPtr SearchAddIndex (SearchIndexListPtr list,  
char* fullPath, ASUns32 flags);
```

### Description

Adds a specified index to the shelf.

**NOTE:** Superceded in Acrobat 6.0 by [SearchAddIndexEx](#).

### Parameters

---

<b>list</b>	The list of indices on the shelf, obtained using <a href="#">SearchGetIndexList</a> .
<b>fullPath</b>	A platform-dependent path to the index. <ul style="list-style-type: none"><li>● On the Mac OS, it is of the form <code>MyDisk:TopFolder:BottomFolder:Strange.pdx</code></li><li>● In Windows, it is of the form <code>C:\LVL1\MYFILES\INDEX</code>.</li></ul> In UNIX and Windows, the <code>.PDX</code> extension is automatically added to the specified pathname.
<b>flags</b>	Flags that indicate the state of the index. Must be an OR of the following values (see <code>SrchType.h</code> ). <ul style="list-style-type: none"><li>● <b>kIndexAvailableFlag</b> — The index is available for searching. Indices that are not available appear grayed out in the Search plug-in's user interface.</li><li>● <b>kIndexSelectedFlag</b> — The index is used for searching. Indices that are selected appear with a filled in checkbox in the Search plug-in's user interface.</li><li>● <b>kIndexPtrInvalidFlag</b> — (Read only—cannot be set) The index cannot be located; it does not exist.</li></ul> In practice, <b>kIndexAvailableFlag</b> should always be set.

---

### Return Value

The index that was added. This value may be used in subsequent calls to remove the index, obtain its title, and so forth.

Returns **kAddIndexFailed** if the specified index could not be added to the shelf.

Returns **kIndexExists** if the index is already on the shelf.

## SearchAddIndexEx

```
SearchIndexPtr SearchAddIndexEx (SearchIndexListPtr list,
ASFileSys fileSys, ASPathName indexPath, ASUns32 flags);
```

### Description

Adds a specified index to the shelf, using the Acrobat 6.0 file system and path mechanism. See the *Acrobat Core API Reference* for information on the **ASFileSys** and **ASPathName** structures.

**NOTE:** Supersedes **SearchAddIndex** in Acrobat 6.0.

### Parameters

<b>list</b>	The list of indices on the shelf, obtained using <b>SearchGetIndexList</b> .
<b>fileSys</b>	The file system on which the index is found.
<b>indexPath</b>	The path to the index.
<b>flags</b>	<p>Flags that indicate the state of the index. Must be an OR of the following values (see <b>SrchType.h</b>).</p> <ul style="list-style-type: none"> <li>● <b>kIndexAvailableFlag</b> — The index is available for searching. Indices that are not available appear grayed out in the Search plug-in's user interface.</li> <li>● <b>kIndexSelectedFlag</b> — The index is used for searching. Indices that are selected appear with a filled in checkbox in the Search plug-in's user interface.</li> <li>● <b>kIndexPtrInvalidFlag</b> — (Read only—cannot be set) The index cannot be located; it does not exist.</li> </ul> <p>In practice, <b>kIndexAvailableFlag</b> should always be set.</p>

### Return Value

The index that was added. This value may be used in subsequent calls to remove the index, obtain its title, and so forth.

Returns **kAddIndexFailed** if the specified index could not be added to the shelf.  
Returns **kIndexExists** if the index is already on the shelf.



---

## SearchCountIndexList

```
ASInt32 SearchCountIndexList (SearchIndexListPtr list);
```

### Description

Gets the number of indices currently on the shelf.

### Parameters

---

<code>list</code>	The list of indices on the shelf, obtained using <a href="#">SearchGetIndexList</a> .
-------------------	---

---

### Return Value

The number of indices on the shelf.

## SearchExecuteQuery

```
ASBool SearchExecuteQuery (char* queryStr, ASInt16 parserID,
ASInt16 sortCount, char** sortSpec, boolean** sortWays,
ASUns32 wordOptions, ASInt16 overrideWordOptions,
ASInt16 maxDocs);
```

### Description

Executes a specified query, using the set of indices currently on the shelf. The search results are displayed in the Acrobat Search plug-in's Results window.

**NOTE:** Superseded in Acrobat 6.0 by [SearchExecuteQueryEx](#), which allows a broader range of search locations.

### Parameters

<b>queryStr</b>	The query. Its format is the same as what a user would type into the Search query dialog, and depends on the search language specified by <b>parserID</b> .
<b>parserID</b>	The search language used in the query. Must be one of the following values (see <code>SrchType.h</code> ). <ul style="list-style-type: none"> <li>● <b>kParserSimple</b> — Allows only simple phrase searches; does not allow boolean searching.</li> <li>● <b>kParserCQL</b> — Allows boolean searches using <b>AND</b>, <b>OR</b>, and <b>NOT</b>, as described in the Acrobat Search plug-in's online help file.</li> <li>● <b>kParserBPlus</b> — The Verity BooleanPlus query language. Contact Verity for further information on this language.</li> </ul>
<b>sortCount</b>	The number of entries in the <b>sortSpec</b> list.
<b>sortSpec</b>	An array of strings, each specifying a key whose value is used to sort the search results. The first entry is the primary sort key, the second is the secondary sort key, and so forth. Each string may be any field that appears in the index, plus <b>Score</b> (which sorts results by relevance ranking). Some common fields are <b>Title</b> , <b>ModificationDate</b> , <b>CreationDate</b> , and <b>Keywords</b> .
<b>sortWays</b>	An array of booleans indicating sort order, corresponding to the array of <b>sortSpecs</b> with <b>true</b> = ascending, <b>false</b> = descending.

<b>wordOptions</b>	<p>Search options. Must be an <b>OR</b> of the following values (see <code>SrchType.h</code>).</p> <ul style="list-style-type: none"> <li>● <b>kWordOptionCase</b> — The search is case-sensitive.</li> <li>● <b>kWordOptionStemming</b> — Find not only the specified word, but other words that have the same stem (for example, run and ran have the same stem).</li> <li>● <b>kWordOptionSoundsLike</b> — Find not only the specified word, but other words that sound like it.</li> <li>● <b>kWordOptionThesaurus</b> — Find not only the specified word, but other words that have the same meaning.</li> <li>● <b>kWordOptionProximity</b> — Consider the proximity of results when using the <b>AND</b> operator to look for more than one word in a document. Without this option, <b>ANDed</b> terms can be anywhere in a document. Searching for “red” and “blue,” for example, finds a document where “red” is the first word on the first page and where “blue” is the last word on the last page. With this option, however, <b>ANDed</b> terms must be within two or three pages of each other to be found. Also, the closer <b>ANDed</b> terms appear together, the higher the relevance ranking of the document that contains them.</li> <li>● <b>kWordOptionRefine</b> — Do not search the entire list of indices, but only the documents that matched the previous search. This is used to refine the results of the previous search.</li> </ul> <p>The manner in which <b>wordOptions</b> is used depends on the value of <b>overrideWordOptions</b>.</p>
<b>overrideWordOptions</b>	<p>Flag that indicates whether <b>wordOptions</b> is <b>ORed</b> with the search options set in the user interface, or used instead of them. If 0, <b>wordOptions</b> is <b>ORed</b> with the user interface search options, and the resulting value is used. If non-zero, <b>wordOptions</b> is used instead of the user interface search options.</p>
<b>maxDocs</b>	<p>The maximum number of documents to display in the Results window. If more documents than this have hits, only the first <b>maxDocs</b> are displayed. <b>maxDocs</b> must be no greater than 999.</p>

### Return Value

**true** on success, otherwise **false**.

## SearchExecuteQueryEx

```
ASBool SearchExecuteQueryEx
(const SearchQueryDataRec *queryData);
```

### Description

Executes a specified query, using the given set of search parameters. The search results are displayed in the Acrobat Search plug-in's Results window.

**NOTE:** Supersedes [SearchExecuteQuery](#) in Acrobat 6.0.

### Parameters

<code>queryData</code>	A pointer to the structure containing the search parameters.
------------------------	--

### Return Value

`true` if successful, `false` otherwise.

### Data Structure

```
typedef struct _t_SearchQueryDataRec {
    ASSize_t size;
    ASText query;
    SearchType type;
    SearchMatchOption match;
    SearchWordOptions options;
    SearchScope scope;
    ASPathName path;
    ASFileSys fs;
    ASUns16 maxDocs;
} SearchQueryDataRec;
```

<code>size</code>	Size of the data structure. Must be set to <code>sizeof(SearchQueryDataRec)</code> .
<code>query</code>	Text to be searched.
<code>type</code>	Location to search in. One of: <ul style="list-style-type: none"> <li>● <code>kSearchActiveDoc</code></li> <li>● <code>kSearchFolder</code></li> <li>● <code>kSearchIndex</code></li> <li>● <code>kSearchActiveIndexes</code></li> </ul>

<b>match</b>	How the <b>query</b> text should be matched in the document. One of: <ul style="list-style-type: none"> <li>● <b>kMatchPhrase</b></li> <li>● <b>kMatchAllWords</b></li> <li>● <b>kMatchAnyWords</b></li> <li>● <b>kBooleanQuery</b></li> </ul>
<b>options</b>	Search options. A logical OR of the following <b>ASUns32</b> values: <ul style="list-style-type: none"> <li>● <b>kWordOptionCase</b> — The search is case-sensitive.</li> <li>● <b>kWordOptionStemming</b> — Find not only the specified word, but other words that have the same stem (for example, run and ran have the same stem).</li> <li>● <b>kWordOptionSoundsLike</b> — Find not only the specified word, but other words that sound like it.</li> <li>● <b>kWordOptionThesaurus</b> — Find not only the specified word, but other words that have the same meaning.</li> <li>● <b>kWordOptionProximity</b> — Consider the proximity of results when using the <b>AND</b> operator to look for more than one word in a document. Without this option, <b>ANDed</b> terms can be anywhere in a document. Searching for “red” and “blue,” for example, finds a document where “red” is the first word on the first page and where “blue” is the last word on the last page. With this option, however, <b>ANDed</b> terms must be within two or three pages of each other to be found. Also, the closer <b>ANDed</b> terms appear together, the higher the relevance ranking of the document that contains them.</li> <li>● <b>kWordOptionRefine</b> — Do not search the entire list of indices, but only the documents that matched the previous search. This is used to refine the results of the previous search.</li> <li>● <b>kWordOptionWholeWord</b></li> <li>● <b>kWordOptionIgnoreFH</b></li> </ul>
<b>scope</b>	Content in the PDFs that should be searched. A logical OR of the following <b>ASUns32</b> values: <ul style="list-style-type: none"> <li>● <b>kSearchDocumentText</b></li> <li>● <b>kSearchBookmarks</b></li> <li>● <b>kSearchMarkup</b></li> <li>● <b>kSearchDocumentXMP</b></li> <li>● <b>kSearchSignatures</b></li> <li>● <b>kSearchDocInfo</b></li> <li>● <b>kSearchJPEGEExif</b></li> <li>● <b>kSearchEveryWhere</b></li> </ul>
<b>path</b>	The path of the folder or index. Required only when <b>type</b> is <b>kSearchFolder</b> or <b>kSearchIndex</b> .
<b>fs</b>	The <b>ASFileSys</b> of the folder or index. Required only when <b>type</b> is <b>kSearchFolder</b> or <b>kSearchIndex</b> .



---

**maxDocs** The maximum number of documents to display in the Results window. If more documents than this have hits, only the first **maxDocs** are displayed. **maxDocs** must be no greater than 999.

---



---

## SearchGetIndexByPath

```
SearchIndexPtr SearchGetIndexByPath (SearchIndexListPtr list,  
char* fullPath);
```

### Description

Gets the index that has the specified path. The index must already be on the shelf. The index can be passed to other Search plug-in API methods to remove it from the shelf, obtain its title, and so forth.

**NOTE:** Superseded by [SearchGetIndexPathEx](#) in Acrobat 6.0.

### Parameters

---

<b>list</b>	The list of indices on the shelf, obtained using <a href="#">SearchGetIndexList</a> .
<b>fullPath</b>	A platform-dependent path to the index. On the Mac OS, it is of the form <b>MyDisk:TopFolder:BottomFolder:Strange.pdx</b> . In Windows, it is of the form <b>C:\LVL1\MYFILES\INDEX</b> . In Windows, the <b>.PDX</b> extension is automatically added to the specified pathname.

---

### Return Value

The specified index. This value may be used in subsequent calls to remove the index, obtain its title, and so forth.

---

## SearchGetIndexByPathEx

```
SearchIndexPtr SearchGetIndexByPathEx  
(SearchIndexListPtr list, ASFileSys fileSys,  
ASPathName indexPath);
```

### Description

Gets the index that has the specified path. The index must already be on the shelf. The index can be passed to other Search plug-in API methods to remove it from the shelf, obtain its title, and so forth.

**NOTE:** Supersedes [SearchGetIndexPath](#) in Acrobat 6.0.

### Parameters

---

<b>list</b>	The list of indices on the shelf, obtained using <a href="#">SearchGetIndexList</a> .
<b>fileSys</b>	The file system on which the index is found.
<b>indexPath</b>	The path to the index.

---

### Return Value

The specified index. This value may be used in subsequent calls to remove the index, obtain its title, and so forth.

---

## SearchGetIndexFlags

```
ASUns32 SearchGetIndexFlags (SearchIndexPtr index);
```

### Description

Gets the flags for a specified index.

### Parameters

<b>index</b>	The index whose flags are to be obtained (set). The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
<b>flags</b>	<p>Flags that indicate whether the status of the index. <b>flags</b> must be an OR of the following values (see <a href="#">SrchType.h</a>).</p> <ul style="list-style-type: none"> <li>● <b>kIndexAvailableFlag</b> — The index is available for searching. Indices that are not available appear grayed out in the Search plug-in's user interface.</li> <li>● <b>kIndexSelectedFlag</b> — The index is used for searching. Indices that are selected appear with a filled in checkbox in the Search plug-in's user interface.</li> <li>● <b>kIndexPtrInvalidFlag</b> — (Read only—cannot be set) The index cannot be located; it does not exist.</li> </ul> <p>In practice, <b>kIndexAvailableFlag</b> should always be set.</p>

### Return Value

The flags returned are the actual values set, and may not always be the same as the requested value.

## SearchGetIndexList

```
SearchIndexListPtr SearchGetIndexList (void);
```

### Description

Gets a list of the indices currently on the shelf.

### Parameters

None

### Return Value

The list of indices currently on the shelf. This value can subsequently be used by other Search plug-in methods to obtain information about a specific index, the number of indices on the shelf, and so forth.



---

## SearchGetIndexPath

```
char* SearchGetIndexPath (SearchIndexPtr index);
```

### Description

Gets the platform-dependent path for a specified index.

**NOTE:** Superseded in Acrobat 6.0 by [SearchGetIndexPathEx](#).

### Parameters

---

<b>index</b>	The index whose path is obtained. The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
--------------	--

---

### Return Value

A platform-dependent path to the index. On the Mac OS, it is of the form: **MyDisk:TopFolder:BottomFolder:Strange.pdx**. In Windows, it is of the form **C:\LVL1\MYFILES\INDEX**. In Windows, the **.PDX** extension is automatically added to the specified pathname.

---

## SearchGetIndexPathEx

```
ASPathName SearchGetIndexPathEx (SearchIndexPtr index);
```

### Description

Gets the path for a specified index as an **ASPathName** object.

**NOTE:** Supercedes [SearchGetIndexPath](#) in Acrobat 6.0.

### Parameters

---

<b>index</b>	The index whose path is obtained. The index may be obtained using <a href="#">SearchGetIndexPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
--------------	--

---

### Return Value

The **ASPathName** object.



---

## SearchGetIndexFileSys

```
ASFileSys SearchGetIndexFileSys (SearchIndexPtr index);
```

### Description

Gets the file system for a specified index as an **ASFileSys** object.

### Parameters

---

<b>index</b>	The index whose path is obtained. The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
--------------	--

---

### Return Value

The **ASFileSys** object.

---

## SearchGetIndexTitle

```
char* SearchGetIndexTitle (SearchIndexPtr index);
```

### Description

Gets the title of a specified index.

**NOTE:** Superseded in Acrobat 6.0 by [SearchGetIndexTitleEx](#).

### Parameters

---

<b>index</b>	The index whose title is obtained. The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
--------------	---

---

### Return Value

The title of the specified index.



---

## SearchGetIndexTitleEx

```
ASText SearchGetIndexTitleEx (SearchIndexPtr index);
```

### Description

Gets the title of a specified index as an **ASText** object.

**NOTE:** Supercedes [SearchGetIndexTitle](#) in Acrobat 6.0.

### Parameters

---

<b>index</b>	The index whose title is obtained. The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
--------------	---

---

### Return Value

The title of the specified index as an **ASText** object.

---

## SearchGetNthIndex

```
SearchIndexPtr SearchGetNthIndex (SearchIndexListPtr list,  
ASUns32 n);
```

### Description

Gets the  $n^{\text{th}}$  index on the shelf. The index can be passed to other Search plug-in API methods to remove it from the shelf, obtain its title, and so forth.

### Parameters

---

<b>list</b>	The list of indices on the shelf, obtained using <a href="#">SearchGetIndexList</a> .
<b>n</b>	The index to get. The first index on the shelf is index zero.

---

### Return Value

The  $n^{\text{th}}$  index on the shelf. This value may be used in subsequent calls to remove the index, obtain its title, and so forth.



---

## SearchIsLegacySearchAvailable

```
ASBool SearchIsLegacySearchAvailable (void);
```

### Description

Tests whether the search mechanism (Search5) for previous Acrobat versions (prior to 6.0) is available for the current system. When Search5 is available, you can search indexes from Acrobat 5.0 and earlier.

### Parameters

None

### Return Value

**true** if legacy searches are available, **false** otherwise.

---

## SearchRemoveIndex

```
void SearchRemoveIndex (SearchIndexListPtr list,  
SearchIndexPtr index);
```

### Description

Removes the specified index from the shelf.

### Parameters

---

<b>list</b>	The list of indices on the shelf, obtained using <a href="#">SearchGetIndexList</a> .
<b>index</b>	The index to be removed. The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .

---

### Return Value

None

---

## SearchSetIndexFlags

```
ASUns32 SearchSetIndexFlags (SearchIndexPtr index,  
ASUns32 flags);
```

### Description

Sets the flags for a specified index.

### Parameters

---

<b>index</b>	The index whose flags are to be set. The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
<b>flags</b>	Flags that indicate the status of the index. <b>flags</b> must be an OR of the values (see <a href="#">SrchType.h</a> ). <ul style="list-style-type: none"><li>● <b>kIndexAvailableFlag</b> — The index is available for searching. Indices that are not available appear grayed out in the Search plug-in's user interface.</li><li>● <b>kIndexSelectedFlag</b> — The index is used for searching. Indices that are selected appear with a filled in checkbox in the Search plug-in's user interface.</li><li>● <b>kIndexPtrInvalidFlag</b> — (Read only—cannot be set) The index cannot be located; it does not exist.</li></ul> In practice, <b>kIndexAvailableFlag</b> should always be set.

---

### Return Value

The flags returned are the actual values set, and may not always be the same as the requested value.



# Apple Events

---

## SearchAddIndex

### Description

Adds a specified index to the shelf.

### Apple Event ID

`kSearchAddIndex ('addx')`

### Apple Event Parameters

<code>kIndexListTag ('SilP'), typeLongInteger</code>	An opaque <code>void*</code> representing the shelf, obtained from <code>SearchGetIndexList</code> .
<code>kPathTag ('Path'), typeChar</code>	Macintosh full path representing an index, of the form: <code>MyDisk:TopFolder:BottomFolder:Strange.pdx</code>
<code>kFlagTag ('Flag'), typeLongInteger</code>	Index flags. See <a href="#">SearchGetIndexFlags</a> for a description of them. The <code>kIndexAvailable</code> flag should always be set.

### Return Value

`kIndexTag ('SixP'), typeLongInteger`

An opaque `void*` representing an index. Returns `NULL` if failure. Returns

`#define kIndexExists ((SearchIndexPtr)-1)`

if the index already exists in the index list. If the index already exists, you can retrieve it using [SearchGetIndexByPath](#).

---

## SearchCountIndexList

### Description

Gets the number of indices currently on the shelf.

### Apple Event ID

`kSearchCountIndexList ('cidx')`

### Apple Event Parameters

<code>kIndexListTag ('SilP'),</code>	An opaque <code>void*</code> representing the shelf, obtained
<code>typeLongInteger</code>	from <a href="#">SearchGetIndexList</a> .

### Return Value

`kIndexListTag ('SilP'), typeLongInteger`

Number of indices on the shelf (`kIndexListTag` here is not semantically correct, but works).

## SearchDoQuery

### Description

Executes a specified query, using the set of indices currently on the shelf. The search results are displayed in the Acrobat Search plug-in's Results window.

### Apple Event ID

`kSearchDoQuery ('kwry')`

### Apple Event Parameters

<code>kQueryStringTag ('Qury'), typeChar</code>	The query string, a <b>NULL</b> -terminated block of text. Its format is the same as what a user would type into the search Query window, and depends on the search language specified by <code>kParserTag</code> .
<code>kParserTag ('Prsr'), typeShortInteger</code>	The query parser to use; may be one of (see <code>SrchType.h</code> ): <ul style="list-style-type: none"> <li>● <code>kParserSimple 0</code> — Allows only simple phrase searches; does not allow boolean searching.</li> <li>● <code>kParserCQL 1</code> — Allows boolean searches using AND, OR, and NOT, as described in the Acrobat Search plug-in's online help file.</li> <li>● <code>kParserBPlus 2</code> — The Verity BooleanPlus query language. Contact Verity for further information on this language.</li> </ul>
<code>kSortSpecTag ('Sort'), typeAEList</code>	A list of C strings representing fields to sort by. The first element is the first level sort, the second is the second level sort, and so forth. Each string may be any field that appears in the index, plus <b>Score</b> (which sorts results by relevance ranking). Some common fields are Title, ModificationDate, CreationDate, and Keywords.
<code>kWordOptionsTag ('WOpt'), typeLongInteger</code>	A bit field of word options. Must be a logical OR of the values listed below in <a href="#">"Search Plug-in Word Options For Apple Events."</a> The manner in which the options are used depends on the value associated with <code>kOptionsOverrideTag</code> .
<code>kOptionsOverrideTag ('WOer'), typeShortInteger</code>	Flag that indicates whether the word options are OR'ed with the search options set in the user interface, or used instead of them. If 0, the word options are OR'ed with the user interface search options, and the resulting value is used. If non-zero, the word options are used instead of the user interface search options.

**kMaxDocsTag**  
 ( 'MaxD' ) ,  
 typeShortInteger

The maximum number of documents to display in the Results window. If more documents than this have hits, only the first **maxDocs** are displayed. **maxDocs** must be no greater than 999.

**Return Value**

None

**Search Plug-in Word Options For Apple Events**

<b>kWordOptionCase</b>	The search is case-sensitive.
<b>kWordOptionStemming</b>	Find not only the specified word, but other words that have the same stem (for example, run and ran have the same stem).
<b>kWordOptionSoundsLike</b>	Find not only the specified word, but other words that sound like it.
<b>kWordOptionThesaurus</b>	Find not only the specified word, but other words that have the same meaning.
<b>kWordOptionProximity</b>	Consider the proximity of results when using the AND operator to look for more than one word in a document. Without <b>kWordOptionProximity</b> , AND'ed terms can be anywhere in a document. Searching for "red" and "blue," for example, finds a document where "red" is the first word on the first page and where "blue" is the last word on the last page. With <b>kWordOptionProximity</b> , however, AND'ed terms must be within two or three pages of each other to be found. Also, with <b>kWordOptionProximity</b> , the closer AND'ed terms appear together, the higher the relevance ranking of the document that contains them.
<b>kWordOptionRefine</b>	Do not search the entire list of indices, but only the documents that matched the previous search. This is used to refine the results of the previous search.



---

## SearchGetIndexByPath

### Description

Gets the index that has the specified path. The index must already be on the shelf. The index can be passed to other Search Apple events to remove it from the shelf, obtain its title, and so forth.

### Apple Event ID

`kSearchGetIndexByPath ('fpdx')`

### Apple Event Parameters

<code>kIndexListTag ('SilP'), typeLongInteger</code>	An opaque <code>void*</code> representing the shelf, obtained from <a href="#">SearchGetIndexList</a> .
<code>kPathTag ('Path'), typeChar</code>	Macintosh full path representing an index, of the form: <code>MyDisk:TopFolder:BottomFolder:Strange.pdx</code>

### Return Value

`kIndexTag ('SixP'), typeLongInteger`

An opaque `void*` representing an index. Returns `NULL` if the specified index is gone.

---

## SearchGetIndexFlags

### Description

Get the flags for an index.

### Apple Event ID

`kSearchGetIndexFlags` ('gfdx')

### Apple Event Parameters

`kIndexTag` ('SixP'), `void*` An opaque `void*` representing an index.  
`typeLongInteger`

### Return Value

`kFlagTag` ('Flag'), `typeLongInteger`

A logical OR of the following:

`kIndexAvailableFlag` (1L << 0) — Set if the index is available for searching.

`kIndexSelectedFlag` (1L << 1) — Set if the index appears with a check mark in the Search plug-in's user interface.

`kIndexPtrInvalidFlag` (1L << 31) — Set if the index in is not valid or is no longer valid.



---

## SearchGetIndexList

### Description

Gets a list of the indices currently on the shelf.

### Apple Event ID

`kSearchGetIndexList ('gidx')`

### Apple Event Parameters

None

### Return Value

`kIndexListTag ('SilP'), typeLongInteger`

An opaque `void*` representing the list of indices currently on the shelf. This value can subsequently be used by other search Apple events to obtain information about a specific index, the number of indices on the shelf, and so forth.

---

## SearchGetIndexPath

### Description

Gets the full path to an index.

### Apple Event ID

`kSearchGetIndexPath ('gpdx')`

### Apple Event Parameters

<code>kIndexTag ('SixP'), typeLongInteger</code>	An opaque <code>void*</code> representing the index whose path is to be obtained. The index may be obtained using <a href="#">SearchGetIndexPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
--	--

### Return Value

`kPathTag ('Path'), typeChar`

A `NULL`-terminated character string representing the full path of the index. Returns an empty string if the requested index is not valid.



---

## SearchGetIndexTitle

### Description

Gets the title of an index.

### Apple Event ID

`kSearchGetIndexTitle ('gtdx')`

### Apple Event Parameters

<code>kIndexTag ('SixP'), typeLongInteger</code>	An opaque <code>void*</code> representing the index whose title is to be obtained. The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .
--	---

### Return Value

`kTitleTag ('Titl'), typeChar`

A `NULL`-terminated character string representing the title of the index. If there is no title, it will return the index's path. Returns an empty string if the requested index is not valid.

---

## SearchGetNthIndex

### Description

Gets the  $n^{\text{th}}$  index on the shelf. The index can be passed to other Search Apple events to remove it from the shelf, obtain its title, and so forth.

### Apple Event ID

`kSearchGetNthIndex ('fndx')`

### Apple Event Parameters

<code>kIndexListTag ('SilP'), typeLongInteger</code>	An opaque <code>void*</code> representing the shelf, obtained from <a href="#">SearchGetIndexList</a> .
<code>kNthIndexTag ('Enth'), typeLongInteger</code>	The index to get. The first index on the shelf is index zero.

### Return Value

`kIndexTag ('SixP'), typeLongInteger`

An opaque `void*` representing an index. Returns **NULL** if the  $n^{\text{th}}$  index is gone.



---

## SearchRemoveIndex

### Description

Removes the specified index from the shelf.

### Apple Event ID

`kSearchRemoveIndex ('rmdx')`

### Apple Event Parameters

<code>kIndexListTag ('SilP'),</code> <code>typeLongInteger</code>	An opaque <code>void*</code> representing the shelf, obtained from <a href="#">SearchGetIndexList</a> .
<code>kIndexTag ('SixP'),</code> <code>typeLongInteger</code>	An opaque <code>void*</code> representing the index to be removed. The index may be obtained using <a href="#">SearchGetIndexByPath</a> , <a href="#">SearchGetNthIndex</a> , or <a href="#">SearchAddIndex</a> .

### Return Value

None

---

## SearchSetIndexFlags

### Description

Sets the flags for an index.

### Apple Event ID

`kSearchSetIndexFlags` ('sfdx')

### Apple Event Parameters

<code>kIndexTag</code> ('SixP'), <code>typeLongInteger</code>	An opaque <code>void*</code> representing an index.
<code>kFlagTag</code> ('Flag'), <code>typeLongInteger</code>	Index flags. See the description in <a href="#">SearchGetIndexFlags</a> . In practice, <code>kIndexAvailableFlag</code> should always be set.

### Return Value

`kFlagTag` ('Flag'), `typeLongInteger`

Index flags. See the description in [SearchGetIndexFlags](#). This value is returned because it is possible for a request to set a flag to fail.

# DDE Messages

A client can connect to the Search plug-in via DDE using the service name "Acrobat Search" and the topic name "Acrobat Search".

```
DdeInitialize(&id, &DDE_ProcessMessage, APPCMD_CLIENTONLY, 0);
hszServerName = DdeCreateStringHandle(id, "Acrobat Search", 0);
hszTopicName = DdeCreateStringHandle(id, "Acrobat Search", 0);
hConv = DdeConnect(id, hszServerName, hszTopicName, NULL);
```

After a connection has been made, a single poke transaction will submit a search query. Two types of queries are supported: simple query and query.

---

## Simple Query Item

A simple query has the item name "SimpleQuery". When using a simple query, pass only a string that contains the query, using the ASQL query parser's format (see [QLangType\\_CQL](#) in [Table 1, "Query language type constants"](#)). It is not possible to choose another parser or to set word options using the simple query item.

---

## Query Item

Query has the item name "Query". When using query, a **QueryData** structure is passed. This structure contains the query, as well as specifying the query parser to use and additional options.

```
hszItemName = DdeCreateStringHandle(id, "Query", 0);
DdeClientTransaction(qd, nLen, hConv, hszItemName, CF_TEXT, XTYP_POKE,
1000, &dwResult);
DdeDisconnect(hConv)
```

The global data handle (**qd**) passed to the server must be in the following format:

```
typedef struct _QueryData {
    eQLangType qlt;
    boolean bOverrideWordOptions;
    uns32 nWordOptions;
    uns16 nMaxDocs;
    uns16 nQueryOffset;
    uns16 nNumSorts; //deprecated in Acrobat 6.0
    uns16 nSortOffset[QP_MAX_SORT_FIELDS]; //deprecated in Acrobat 6.0
    boolean bSortWays[QP_MAX_SORT_FIELDS]; //deprecated in Acrobat 6.0
```

```
    unsigned char cData[1];
} QueryData;
```

<b>qlt</b>	The query language type. Must be one of the values shown in <a href="#">Table 1, "Query language type constants"</a> .
<b>bOverrideWordOptions</b>	Indicates that the client wishes to use different word options than those currently set by the user.
<b>nWordOptions</b>	The word options. Must be an <b>OR</b> of the values shown in <a href="#">Table 2, "Word option bit-flag constants"</a> .
<b>nMaxDocs</b>	If non-zero, the client wishes to use a different maximum documents limit than the limit currently set by the user.
<b>nSortOffsets</b>	A list of offsets into the <b>cData</b> chunk. Each offset points to a <b>NULL</b> -terminated string containing the field name.  <b>NOTE:</b> This value has no effect in Acrobat 6.0, because sort options are not valid.
<b>nQueryOffset</b>	An offset into the <b>cData</b> chunk that points to a <b>NULL</b> -terminated string containing the query to execute.
<b>nNumSorts</b>	The number of fields in the sort spec. If this number is 0, the plug-in uses the current sort spec set by the user.  <b>NOTE:</b> This value has no effect in Acrobat 6.0, because sort options are not valid.
<b>bSortWays</b>	A list of sort order flags, one for each sort field. <b>true</b> indicates an ascending sort, and <b>false</b> indicates a descending sort.  <b>NOTE:</b> This value has no effect in Acrobat 6.0, because sort options are not valid.

**TABLE 1**      **Query language type constants**

<b>QLangType_Simple</b>	Allows only simple phrase searches; does not allow boolean searching.  <b>NOTE:</b> This query type does not work in the DDE interface of Search plug-in shipped with version 2.0 of Acrobat.
<b>QLangType_CQL</b>	Allows boolean searches using <b>AND</b> , <b>OR</b> , and <b>NOT</b> , as described in the Acrobat Search plug-in's online help file.

**TABLE 1** Query language type constants

<b>QLangType_Passthrough</b>	The Verity BooleanPlus query language. Contact Verity for further information on this language.
------------------------------	---

**TABLE 2** Word option bit-flag constants

<b>QPON_Case</b>	The search is case-sensitive.
<b>QPON_Stemming</b>	Find not only the specified word, but other words that have the same stem (for example, run and ran have the same stem).
<b>QPON_SoundsLike</b>	Find not only the specified word, but other words that sound like it.
<b>QPON_Thesaurus</b>	Find not only the specified word, but other words that have the same meaning.
<b>QPON_Proximity</b>	Consider the proximity of results when using the <b>AND</b> operator to look for more than one word in a document. Without this option, <b>ANDed</b> terms can be anywhere in a document. Searching for "red" and "blue," for example, finds a document where "red" is the first word on the first page and where "blue" is the last word on the last page. With this option, however, <b>ANDed</b> terms must be within two or three pages of each other to be found. Also, the closer <b>ANDed</b> terms appear together, the higher the relevance ranking of the document that contains them.
<b>QPON_Refine</b>	Do not search the entire list of indices, but only the documents that matched the previous search. This is used to refine the results of the previous search.

To create and populate this structure correctly, the client must know the sum of the lengths of each sort field (**s1s**), the length of the query (**lq**), and the size of the **QueryData** structure. The client then allocates memory as follows:

```
nSize = sizeof(QueryData) + s1s + lq;
qd = (QueryData *)malloc(nSize);
```

For example, if the query was “Adobe” and the sort spec was “Title” ascending and “Score” descending then the structure would be packed as follows:

```
memset(qd, 0, nSize);
qd->nQueryOffset = 0;
strcpy(&cData[0], "Adobe");
qd->nNumSort = 2;
qd->nSortOffset[0] = strlen("Adobe") + 1;
qd->bSortWays[0] = TRUE;
strcpy(&cData[qd->nSortOffset[0]], "Title");
qd->bSortWays[1] = FALSE;
qd->nSortOffset[1] = qd->nSortOffset[0] + strlen("Title") + 1;
strcpy(&cData[qd->nSortOffset[1]], "Score");
```

---

## Manipulating Indices Through DDE

After a connection has been made, a single poke transaction can add, delete, add, or remove indices. The item name to use is “Index”.

```
hszItemName = DdeCreateStringHandle(id, "Index", 0);
DdeClientTransaction(qd, nLen, hConv, hszItemName, CF_TEXT, XTYP_POKE,
1000, &dwResult);
DdeDisconnect(hConv);
```

The global data handle (**gd**) passed to the server must be in the following format:

```
typedef struct _IndexData {
    IndexActionType eAction;
    int16 nIndexOffset;
    int16 nTempNameOffset;
    unsigned char cData[1];
} IndexData;
```

<b>eAction</b>	The operation to be performed on the index, and must be one of values listed in <a href="#">Table 3, “Search plug-in index operation selectors for DDE messages”</a> .
<b>nIndexOffset</b>	An offset into the <b>cData</b> chunk that points to a <b>NULL</b> -terminated string containing the <b>.PDX</b> file representing the index.
<b>nTempNameOffset</b>	An offset into <b>cData</b> . It points to a temporary name that is displayed by the Search plug-in when the index is unavailable. This field must specify an offset either to an empty string (\0) or to a non-empty C string.



**TABLE 3** Search plug-in index operation selectors for DDE messages

<b>IndexAction_Add</b>	Adds an index to the shelf.
<b>IndexAction_Remove</b>	Removes an index from the shelf.
<b>IndexAction_Enable</b>	Enables an index on the shelf.
<b>IndexAction_Disable</b>	Disables an index on the shelf.

To create and populate this structure correctly, the client must know the sum of the lengths of the Index (**li**) and Temp names (**lt**) (including **NULL**-terminating characters), and the size of the **IndexData** structure.

The client then allocates memory as follows:

```
nSize = sizeof(IndexData) + li + lt;
id = (IndexData *)malloc(nSize);
```

For example, to add the index `C:\FOO.PDX` to the Search plug-in's shelf:

```
memset(id, 0, nSize);
id->eAction = IndexAction_Add;
id->nIndexOffset = 0;
strcpy(&id->cData[0], "C:\\FOO.PDX");
id->nTempNameOffset = strlen("C:\\FOO.PDX") + 1;
strcpy(&id->cData[id->nTempNameOffset],
"My Favorite Index");
```



## DDE Messages

*Manipulating Indices Through DDE*



# Lists

The Search plug-in adds a new menu, menu items, and toolbar buttons to the Acrobat viewer.

---

## Menu Names

The Search plug-in adds the following menu to the Acrobat viewer.

Menu name	Description
AcroSrch:ToolsSubMenu	Acrobat Search submenu of Edit menu.

---

## Menu Item Names

The Search plug-in adds the following menu items to the Acrobat viewer.

Menu item name	Description
AcroSrch:Query	Displays the Search dialog.
AcroSrch:Indexes	Displays the Index dialog.
AcroSrch:Results	Displays the Results dialog.
AcroSrch:Assist	Displays the Word Assistant dialog.
AcroSrch:Separator	A separator item in the Search tools menu.
AcroSrch:PrevDoc	Goes to the previous document in the hit list.
AcroSrch:PrevHit	Goes to the previous hit in the hit list.
AcroSrch:NextHit	Goes to the next hit in the hit list.
AcroSrch:NextDoc	Goes to the next document in the hit list.

---

## Toolbar Button Names

The Search plug-in adds the following buttons to the Acrobat viewer toolbar

Button name	Description
AcroSrch:Separator	Separator (not visible).
AcroSrch:Query	Displays the Acrobat Search plug-in's query dialog.
AcroSrch:Results	Displays the Acrobat Search plug-in's search results dialog.
AcroSrch:Prev	Goes to the previous hit in the Acrobat Search plug-in's results list.
AcroSrch:Next	Goes to the next hit in the Acrobat Search plug-in's results list.