



# Appendix A element & element:a/v Web Services Guide

Version 2.0



# Web Services



*This web service documentation assumes that the programmer has a basic knowledge of web services and is using a high level programming language that allows an object model representation of the web service function calls. If user is not using a sufficient programming language or familiar with Web Services, please contact your System Administrator.*

To access the full Web Services Description Language (WSDL) navigate your internet browser to:

<http://<ipaddress>/service/element.asmx?WSDL>

*(where <ipaddress> is the IP address assigned to your Element appliance)*

element

Provides basic access to element TM search engine functionality

The following operations are supported. For a formal definition, please review the [Service Description](#).

- ◆ [Search](#)  
Performs standard search. Requires session.
- ◆ [SearchAll](#)  
Performs standard search on all contexts and schemas. Requires session.
- ◆ [SetResultViewFilter](#)  
Sets filter for results using grouping ID and sorting method. Requires session.
- ◆ [GetSchemaContexts](#)  
Obtains all available contexts used in specified schema
- ◆ [GetResultDocument](#)  
Returns found document. Index must be in range from 0 to amount of found documents - 1. Requires session.
- ◆ [GetAvailableSchemas](#)  
Obtains all available schemas
- ◆ [GetDocumentInfo](#)  
Gets information about document using its ID.
- ◆ [GetResultViewDocument](#)  
Returns found document with account of applied filter. Index must be in range from 0 to amount of filtered documents - 1. Requires session.
- ◆ [GetGroupCategories](#)  
Returns possible grouping. Requires session.
- ◆ [SearchDocuments](#)  
Searches only document. Returns array of document URI.
- ◆ [GetAvailableContexts](#)  
Obtains all available contexts
- ◆ [GetEsxFile](#)  
Gets the esx document that is stored internally.

element WSDL Main Page

## Test access to Element appliance

- Open internet browser
- Navigate to the Element webservice description page located at <http://<ipaddress>/service/element.asmx>  
*where <ipaddress> is the IP address of your Element appliance.*
- The description page gives you a link to the WSDL service description along with links to individual descriptions of the webservice functions exposed by the appliance

## Schemas and Contexts

The Element search engine allows you to narrow your search by specifying schemas (topics) and contexts (sub-topics) to search within.

### Schemas

To discover the schemas available for searching call the webservice function:

#### [GetAvailableSchemas\(\)](#)

The following is a sample SOAP request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /service/element.asmx HTTP/1.1
Host: element.docsoft.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.docsoft.com/services/element/GetAvailableSchemas"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAvailableSchemas xmlns="http://www.docsoft.com/services/element/" />
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAvailableSchemasResponse xmlns="http://www.docsoft.com/services/element/">
      <GetAvailableSchemasResult>
        <SchemaInfo>
          <Name>string</Name>
          <ID>int</ID>
        </SchemaInfo>
        <SchemaInfo>
          <Name>string</Name>
          <ID>int</ID>
        </SchemaInfo>
      </GetAvailableSchemasResult>
    </GetAvailableSchemasResponse>
  </soap:Body>
</soap:Envelope>
```

The function returns an array of schema objects that specify the name and ID of the schema. The available schemas can be displayed to the user or can be hard coded into the page depending on your needs. The schema IDs can be passed to the [Search\(\)](#) webservice function.



# Web Services

## Contexts

To discover the contexts available for a specific schema call the webservice function: **GetSchemaContexts(int schemaId)** passing the ID for the schema whose contexts you wish to discover. The function returns an array of context objects that specify the name and ID of the context. The available contexts can be displayed to the user or can be hard coded into the page depending on your needs. The contexts IDs can be passed to the **Search()** webservice function.

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /service/element.asmx HTTP/1.1
Host: element.docsoft.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.docsoft.com/services/element/GetAvailableContexts"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAvailableContexts xmlns="http://www.docsoft.com/services/element/" />
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetAvailableContextsResponse xmlns="http://www.docsoft.com/services/element/">
      <GetAvailableContextsResult>
        <ContextInfo>
          <Name>string</Name>
          <ID>int</ID>
        </ContextInfo>
        <ContextInfo>
          <Name>string</Name>
          <ID>int</ID>
        </ContextInfo>
      </GetAvailableContextsResult>
    </GetAvailableContextsResponse>
  </soap:Body>
</soap:Envelope>
```

## Searching

### *Simple Search*

Call the function **SearchAll(string query)** using the query input by the user. The function returns a **SearchResult** object with the details of your search results.

### *Complex Search*

Create a Query object. Set the query string property. Set the schema IDs property to specify which schemas should be searched. This property is an array of integers (**int[]**). Set the context IDs property to specify which contexts should be searched. This property is an array of integers (**int[]**). Call the function **Search(Query)** passing the query object you created.

### *Processing search results*

Both the **Search()** and **SearchAll()** webservice functions return a **SearchResult** object: **SearchResult.Count**. Use the count to determine how many documents to display. Call: **GetDocumentResult(int index)** to obtain document results.

### *SearchResult.Statements*

Basic Statements are used convey information about the search that should be displayed to the user. **DidYouMeanStatements** are used as suggestions as to possible misspellings in the search query. They can be used to refine the search.

### *Filtering search results*

Call **GetGroupCategories()** to return a list of all available group categories. An example of a group category would be “**Topic**” and “**File Type**”.

Each group category contains an array of group information objects that describe each individual type of document in the group category. Example: The group category “**File Type**” might have “**XML Document**” and “**Adobe Acrobat Document**” as group information objects.

Call **SetResultViewFilter(int groupId, ResultSort sortMethod)** to set the filter for your results. The **groupId** parameter specifies which group information you want to see. The **ResultSort** object encapsulates instructions for sorting such as by file name, rank, etc. The function returns an integer specifying the number of documents returned after applying the filter.

Call **GetResultViewDocument(int index)** to retrieve individual documents after applying the filter. The index parameter must be between zero and one less than the integer returned from **SetResultViewFilter()**.