

Coveo Enterprise Search 6.0

Push API

The Push API allows users to define a source for which documents are programmatically pushed in the index. The file information is given to the Push API in XML format. It is useful when:

- A custom data repository is used (e.g. a database or in-house application).
- Indexing intervals are controlled.
- Access to the data structure of the documents is granted.

Source Configuration

This is one of the simplest Source type to configure, as the only step to perform is to create a new **Manual** type source. Afterwards, indexing is done using API calls.

Required API Calls

To push a document, perform the following procedure:

1. Open an Admin connection.
2. Locate the source in which to push the documents.
3. Use the *PushDocument* or *PushDocumentEx* method for batchnig.

PushDocument()	Type	Description
p_Document	String	XML structure of the document
p_Commit	Boolean	Forces CES to apply the transaction to the index immediately.

PushDocumentEx()	Type	Description
p_Document	String	XML structure of the document
p_Commit	Boolean	Forces CES to apply the transaction to the index immediately
P_pFailed	String	XML structure of the document whose indexing process has failed. Refer to the Appendix 2 – XSD Schema of the XML of Failed Documents .

Note: On the CES server, it is possible to script call to CES using VBScript. Any language using COM Automation can be used. Refer to the [Appendix 3 – API Call Sample \(VBScript\)](#).

XML Document Structure

The following displays the basic XML structure of the document to send to the API call:

```
<?xml version="1.0">
<CESDocuments>
  <Document>
    <FileName></FileName>
    <FileExtension></FileExtension>
    <ContentType></ContentType>
    <DocumentURI></DocumentURI>
    <PrintableURI></PrintableURI>
    <ClickableURI></ClickableURI>
    <URICaption></URICaption>
    <ModifiedDate></ModifiedDate>
    <CopyProtected></CopyProtected>
    <Content></Content>
    <DocumentPath></DocumentPath>
    <Permission>
      <User></User>
      <Server></Server>
      <Type></Type>
      <Access></Access>
    </Permission>
    <MetaData>
      <Name></Name>
      <Value></Value>
    </MetaData>
  </Document>
</CESDocuments>
```

XML Document Elements

Element Name	Description	Required	Default Value
CESDocuments	Root element of the XML file.	Yes	
Document	Repeated for each document contained in this XML file.	Yes	
FileName	Filename of the document	Yes	
FileExtension	Extension of the document (.txt, .doc, .pdf, etc.) used for content detection.	No	Extracted from the document filename.
ContentType	Mime type of the content which can be indexed (text/plain, text/html, etc.) used for content detection.		
DocumentURI	Unique URI that identifies the document.	Yes	
PrintableURI	URI displayed instead of DocumentURI in the search results.	No	DocumentURI
ClickableURI	URI used as the clickable URI instead of the DocumentURI.	No	DocumentURI
URICaption	Caption that identifies the document URI.	No	
Title	Title of the document.	No	
ModifiedDate	Date at which the document is modified.	No	Indexed date
CopyProtected	Indicates if the document is copy protected.	No	False

Content	Textual content which can be indexed.	No
DocumentPath	Path from which the binary document is loaded.	
BinaryContent	Base64 encoded binary document.	No
Permission	Repeated for each permission specified on a document (at least one Allowed permission must be specified on the document).	Yes
User	User/Group identification.	Yes
Server	Domain/Server on which the user exists.	No
Type	Five types of security can be specified (required): <ul style="list-style-type: none"> • NTUser: Indicate an NT security (user or group). In that case the user identification must be a SID or domain\name (domain\anyone). • CustomGroup: Indicate a custom security mapping. In that case the group must match a custom mapping. • CustomUser: Indicate a custom security mapping. In that case the user must match a custom mapping. • ExternalGroup: Indicate a group that should be expanded by an External Security Provider. • ExternalUser: Indicate a user that should be resolved by an External Security Provider. 	Yes
Access	Indicate if the document is Allowed or Denied for the user (required).	Yes
Metadata	The metadata is copied to CES fields when matching can be established. Repeat to add multiple metadata.	No
Name	Name that identifies the metadata	Yes
Value	Value of the metadata	Yes

FAQ

▶ How do I manage document-level security?

Document-level security is managed by specifying *Allowed/Denied* permissions within each XML document, with the use of Permission elements.

▶ Why isn't there a file level "Permission" menu as on all other source types?

The file level "Permission" menu on the source is created to allow the user to modify the security retrieved by the connectors. In the case of the Push API, no securities are retrieved by the connector; they are specified by the user.

▶ How can I delete a document?

Use the DeleteURI() method found in a source.

▶ What date format can I use for ModifiedDate?

The following date formats are supported:

- "%Y/%m/%d %H:%M:%S" for "2002/01/31 11:53:12"
- "%D %o %y %H:%M:%S %G" for "1 Sep 85 09:54:48 GMT"
- "XXX, %D %o %Y %H:%M:%S %G" for "Sun, 6 Nov 1994 08:49:37 GMT"
- "XXXXXX, %d-%o-%y %H:%M:%S %G" for "Sunday, 06-Nov-94 08:49:37 GMT"
- "XXX %o %d %H:%M:%S %Y" for "Sun Nov 31 08:49:37 1994"

▶ Which elements of the XML structure are optional (what are the default values)?

Refer to the [Appendix 1 – XSD Schema of the XML Document](#). All elements with the required attribute are required, all others are optional.

Note: No default values exist, except for FileExtension, PrintableURI, ClickableUR, ModifiedDate, and CopyProtected.

Appendix 1 – XSD Schema of the XML Document

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

    <!--
    //*****
    // Enumeration
    //*****
    --
    <xs:simpleType name="PermissionTypeEnumeration"
        <xs:restriction base="xs:string"
            <xs:enumeration value="NTUser" /
            <xs:enumeration value="CustomGroup" /
            <xs:enumeration value="CustomUser" /
            <xs:enumeration value="ExternalGroup" /
            <xs:enumeration value="ExternalUser" /
        </xs:restriction
    </xs:simpleType

    <xs:simpleType name="PermissionAccessEnumeration"
```

```

    <xs:restriction base="xs:string"
      <xs:enumeration value="Allowed" /
      <xs:enumeration value="Denied" /
    </xs:restriction
  </xs:simpleType

  <!--
  //*****
  // Schema
  //*****
  --
  <element name="CESDocuments"
    <sequence name="Document"
      <!-- Document identification elements --
      <element name="FileName" type="xs:string" required="true" description="Filename
of the document"/
      <element name="FileExtension" type="xs:string" required="false"
description="Extension of the document (.txt, .doc, .pdg, etc.)"/
      <element name="ContentType" type="xs:string" required="false" description="Mime
type of the indexable content (text/plain, text/html, etc.)"/
      <element name="DocumentURI" type="xs:string" required="true"
description="Unique URI that identify the document"/
      <element name="PrintableURI" type="xs:string" required="false" description="The
URI that is displayed instead of the DocumentURI in the search results"/
      <element name="ClickableURI" type="xs:string" required="false" description="The
URI that is used as the clickable URI instead of the DocumentURI"/

      <!-- Document content elements --
      <!-- Only one content can be specified at a time ether "BinaryContent",
-->
      <!-- "Content" or "DocumentPath":
-->
      <!-- BinaryContent - Should be used to index a binary document. The passed
-->
      <!-- document should be Base64 encoded.
-->
      <!-- Content - Should be used to index only textual content
-->
      <!-- DocumentPath - File path where the document should be load from the
disk-->
      <element name="URICaption" type="xs:string" required="false"
description="Caption that identify the document URI"/
      <element name="Title" type="xs:string" required="false" description="Title of
the document"/
      <element name="ModifiedDate" type="xs:dateTime" required="false"
description="Time at which the document was modified"/
      <element name="CopyProtected" type="xs:boolean" required="false"
description="Indicate if the document is copy protected"/
      <element name="BinaryContent" type="xs:string" required="false"
description="Base64 encoded binary document"/
      <element name="Content" type="xs:string" required="false"
description="Indexable content"/
      <element name="DocumentPath" type="xs:string" required="false"
description="Path from which the binary document is loaded" /

      <!-- Document permissions --
      <sequence name="Permission" required="true"
-->
        <!-- The elements "Type" and "Access" need to be set for all types of
-->
        <!-- securities.
-->
        <!---
-->

```

```

-->         <!-- 5 Types of securities can be specified:
-->
-->         <!--      NTUser      - Indicate an NT security, in such a case the
-->
-->         <!--
-->         0...) -->
-->         <!--
-->         User/Group identification must be an SID (S-1-1-
-->
-->         <!--
-->         or a domain\name (coveo\everyone).
-->
-->         <!--      CustomGroup - Indicate a CustomGroup security mapping.  In such a
-->
-->         <!--
-->         case the "User", "Server" and "CustomMapping"
-->
-->         <!--
-->         elements should be used to identify the custom
--> group. -->
-->         <!--      CustomUser  - Indicate a CustomUser security mapping.  In such a
-->
-->         <!--
-->         case the "User", "Server" and "CustomMapping"
-->
-->         <!--
-->         elements should be used to identify the custom
--> user. -->
-->         <!--      ExternalGroup - Indicate a CustomGroup security mapping.  In such a
-->
-->         <!--
-->         case the "User", and "SecurityProvider" elements
-->
-->         <!--
-->         should be used to identify the external group.
-->
-->         <!--      ExternalUser - Indicate a CustomUser security mapping.  In such a
-->
-->         <!--
-->         case the "User", and "SecurityProvider" elements
-->
-->         <!--
-->         should be used to identify the external user.
-->
-->         <element name="User" type="xs:string" required="true"
description="User/Group identification"/
-->         <element name="Server" type="xs:string" required="false"
description="Domain/Server on which the User exists"/
-->         <element name="CustomMapping" type="xs:string" required="false"
description="Identifier of the Custom Mapping"/
-->         <element name="Type" type="PermissionTypeEnumeration" required="true"
description="Indicate the Type of permission"/
-->         <element name="Access" type="PermissionAccessEnumeration" required="true"
description="Indicate the permission Access type"/>
-->         <element name="SecurityProvider" type="xs:string" required="false"
description="External security provider's name"/>
-->         </sequence>
-->
-->         <!-- Collection of MetaData associated to the document.  The MetaData will
-->
-->         <!-- be binded into CES fields when matching could be established
-->
-->         <sequence name="MetaData">
-->         <element name="Name" type="xs:string" required="true" description="Name
that identify the meta data"/>
-->         <element name="Value" type="xs:string" required="true" description="Value
of the meta data"/>
-->         </sequence>
-->
-->         <!-- Additional data stream that can be binded to the document. -->
-->         <!-- To be valid, the data stream must have a "Name" and a leats on of the
-->
-->         <!-- following Data field must be supplied:
-->

```

```

-->      <!-- BinaryData - Base64 encoded data.
-->
-->      <!-- Data      - Textual data.
-->
-->      <!-- DataPath  - Path on a file from where the data could be loaded.
-->
-->      <sequence name="DataStream">
-->          <element name="Name" type="xs:string" required="true" description="Name
that identify the data stream"/>
-->          <element name="BinaryData" type="xs:string" required="false"
description="Base64 encoded binary data"/>
-->          <element name="Data" type="xs:string" required="false"
description="Indexable textual data"/>
-->          <element name="DataPath" type="xs:string" required="false"
description="Path from which the binary data is loaded" />
-->      </sequence>
-->  </sequence>
--> </element>
--> </xs:schema>
  
```

Appendix 2 – XSD Schema of the XML of Failed Documents

```

<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!--
  //*****
  // PushDocumentEx() result schema.
  //*****
  -->
  <element name="CESFailedDocuments">
    <sequence name="FailedDocument">
      <!-- Error about the document(s) that failed to be indexed -->
      <!-- The possible ErrorIDs are (see CESAdmin.tlb): -->
      <!-- - 1 : A specified permission is invalid. -->
      <!-- - 2 : The "DocumentURI" is invalid. -->
      <!-- - 3 : The "ClickableURI" is invalid. -->
      <!-- - 4 : The "PrintableURI" is invalid. -->
      <!-- - 5 : No document content is specified. -->
      <!-- - 6 : The "DocumentURI" element is missing from a document. -->
      <!-- - 7 : No "CustomMapping" specified for a custom permission. -->
      <!-- - 8 : The "FileName" element is missing. -->
      <!-- - 9 : No permissions specified. -->
      <!-- - 10 : No "SecurityProvider" given for an external permission. -->
      <!-- - 11 : Document too large. -->
      <!-- - 1000 : An unexpected error has occurred. -->
      <element name="ErrorID" type="xs:int" required="true" description="ID that
identify the error that occurred."/>
      <element name="ErrorMsg" type="xs:string" required="true" description="Message
that indicate why the document processing has failed."/>
      <!-- Information about the failed document (not present if the "DocumentURI"
-->
-->      <!-- element was missing from the document).
-->
-->      <element name="Document" required="false">
-->          <!-- This element represent the XML document that had failed to be
processed -->
-->          <!-- by PushDocumentEx(). See the schema "CESXmlDocumentSchema.xsd" for a
-->
-->          <!-- complete description of the document.
-->
-->      </element>
  
```

```

    </sequence>
  </element>
</xs:schema>

```

Appendix 3 – API Call Sample (VBScript)

```
Option Explicit
```

```

Dim Admin: Set Admin = CreateObject("CESAdmin.Admin.4")
Dim P: Set P = Admin.PhysicalIndexes(1)
Dim C: Set C = P.Collections.ItemByName("MyCollection")
Dim S: Set S = C.Sources.ItemByName("MySource")

Dim ModifiedDate: ModifiedDate = Now()

Dim Document: Document = _
  "<?xml version='1.0'>" &_
  "<CESDocuments>" &_
  "  <Document>" &_
  "    <FileName>TestDocument.txt</FileName>" &_
  "    <FileExtension>.txt</FileExtension>" &_
  "    <ContentType></ContentType>" &_
  "    <DocumentURI>my://myuniqueURI</DocumentURI>" &_
  "    <PrintableURI></PrintableURI>" &_
  "    <ClickableURI>http://www.coveo.com</ClickableURI>" &_
  "    <URICaption></URICaption>" &_
  "    <ModifiedDate>" & CStr(ModifiedDate) & </ModifiedDate>" &_
  "    <CopyProtected></CopyProtected>" &_
  "    <Content></Content>" &_
  "    <DocumentPath></DocumentPath>" &_
  "    <Permission>" &_
  "      <User></User>" &_
  "      <Server></Server>" &_
  "      <Type></Type>" &_
  "      <Access></Access>" &_
  "    </Permission>" &_
  "    <MetaData>" &_
  "      <Name></Name>" &_
  "      <Value></Value>" &_
  "    </MetaData>" &_
  "  </Document>" &_
  "</CESDocuments>"

Call S.PushDocument(Document)

```