

# [MS-FSPSCFG]: Processor Server Configuration File Format

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
11/06/2009	0.1	Major	Initial Availability
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.05	Major	Significantly changed the technical content.
12/17/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.6	Minor	Clarified the meaning of the technical content.
04/11/2012	1.6	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.6	No change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>6</b>
1.1 Glossary	6
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Structure Overview (Synopsis)	7
1.4 Relationship to Protocols and Other Structures	7
1.5 Applicability Statement	8
1.6 Versioning and Localization	8
1.7 Vendor-Extensible Fields	8
<b>2 Structures</b>	<b>9</b>
2.1 user_converter_rules.xml	9
2.1.1 Part Enumerations	9
2.1.2 Extensions	10
2.1.3 Global Elements	10
2.1.4 Global Attributes	10
2.1.5 Complex Types	10
2.1.5.1 CT_IFilter	10
2.1.5.2 CT_Trust	10
2.1.5.3 CT_Ext	11
2.1.5.4 CT_MimeMapping	11
2.1.5.5 CT_Mime	11
2.1.6 Simple Types	11
2.1.6.1 ST_Mime	11
2.1.6.2 ST_MimeType	11
2.1.6.3 ST_Name	12
2.2 ManagedProperties.xml	12
2.2.1 Part Enumerations	12
2.2.2 Extensions	12
2.2.3 Global Elements	12
2.2.4 Global Attributes	13
2.2.5 Complex Types	13
2.2.5.1 CT_select	13
2.2.5.2 CT_merge	13
2.2.5.3 CT_property	13
2.2.6 Simple Types	14
2.2.6.1 ST_name	14
2.2.6.2 ST_cpname	14
2.2.6.3 ST_type	14
2.3 PropertyCategories.xml	14
2.3.1 Part Enumerations	15
2.3.2 Extensions	15
2.3.3 Global Elements	15
2.3.4 Global Attributes	15
2.3.5 Complex Types	15
2.3.5.1 CT_category	15
2.3.5.2 CT_indexedproperties	15
2.3.5.3 CT_property	16
2.3.5.4 CT_propset	16

2.3.6	Simple Types .....	16
2.3.6.1	ST_name .....	16
2.3.6.2	ST_indexed .....	16
2.3.6.3	ST_cpname .....	17
2.3.6.4	ST_cpindexed .....	17
2.3.6.5	ST_discover .....	17
2.3.6.6	ST_propset .....	18
2.4	OptionalProcessing.xml .....	18
2.4.1	Part Enumerations .....	18
2.4.2	Extensions .....	18
2.4.3	Global Elements .....	18
2.4.4	Global Attributes .....	19
2.4.5	Complex Types .....	19
2.4.5.1	CT_processor .....	19
2.4.6	Simple Types .....	19
2.4.6.1	ST_name .....	19
2.4.6.2	ST_active .....	19
2.5	XMLMapper.xml .....	20
2.5.1	Part Enumerations .....	20
2.5.2	Extensions .....	20
2.5.3	Global Elements .....	20
2.5.4	Global Attributes .....	21
2.5.5	Complex Types .....	21
2.5.5.1	CT_XMLMappings .....	21
2.5.5.2	CT_Namespace .....	21
2.5.5.3	CT_SubTree .....	21
2.5.5.4	CT_Mapping .....	22
2.5.5.5	CT_MappingGroup .....	22
2.5.6	Simple Types .....	23
2.5.6.1	ST_propset .....	23
2.5.6.2	ST_paragraph-sep .....	23
2.5.6.3	ST_name .....	23
2.5.6.4	ST_uri .....	24
2.5.6.5	ST_path .....	24
2.5.6.6	ST_attr .....	24
2.5.6.7	ST_m-propset .....	24
2.5.6.8	ST_m-type .....	24
2.5.6.9	ST_sep-str .....	25
2.5.6.10	ST_pre-str .....	25
2.5.6.11	ST_post-str .....	25
2.5.6.12	ST_ignore-whitespace .....	25
2.5.6.13	ST_strip-tags .....	26
2.5.6.14	ST_shallow .....	26
2.5.6.15	ST_mode .....	27
2.5.6.16	ST_base-path .....	27
2.5.6.17	ST_mg-base-path .....	27
2.5.6.18	ST_mg-rec-sep-str .....	28
2.5.6.19	ST_mg-rec-pre-str .....	28
2.5.6.20	ST_mg-rec-post-str .....	28
2.5.6.21	ST_mg-select .....	28
2.5.6.22	ST_mg-mode .....	29
2.5.6.23	ST_type .....	29
2.6	CustomPropertyExtractors.xml .....	30

2.6.1	Part Enumerations .....	30
2.6.2	Extensions .....	30
2.6.3	Global Elements .....	30
2.6.4	Global Attributes .....	30
2.6.5	Complex Types .....	30
2.6.5.1	CT_extractor .....	30
2.6.5.2	CT_dictionary .....	31
2.6.6	Simple Types .....	31
2.6.6.1	ST_name .....	31
2.6.6.2	ST_type .....	31
2.6.6.3	ST_property .....	32
2.6.6.4	ST_dictionary_name .....	32
2.6.6.5	ST_yield_values .....	32
<b>3</b>	<b>Structure Examples .....</b>	<b>33</b>
3.1	ManagedProperties.xml .....	33
3.2	PropertyCategories.xml .....	33
3.3	user_converter_rules.xml .....	34
3.4	OptionalProcessing.xml .....	34
3.5	XMLMapper.xml .....	34
3.6	CustomPropertyExtractors.xml .....	35
<b>4</b>	<b>Security Considerations.....</b>	<b>36</b>
<b>5</b>	<b>Appendix A: Full XML Schemas.....</b>	<b>37</b>
5.1	user_converter_rules.xsd.....	37
5.2	ManagedProperties.xsd.....	38
5.3	PropertyCategories.xsd.....	39
5.4	OptionalProcessing.xsd.....	40
5.5	XMLMapper.xsd .....	40
5.6	CustomPropertyExtractors.xsd.....	44
<b>6</b>	<b>Appendix B: Product Behavior .....</b>	<b>46</b>
<b>7</b>	<b>Change Tracking.....</b>	<b>47</b>
<b>8</b>	<b>Index .....</b>	<b>48</b>

# 1 Introduction

This document specifies the Processor Server Configuration File Format Specification. This format specifies file formats that are used to configure the behavior of the Document Processing Service. Two file formats specify how to map between managed properties, crawled properties and crawled property categories. The third file format specifies when to bypass the internal file format detection. The fourth file format contains crawled properties that can be created from specific parts of the crawled items. The last file format specifies whether to enable or disable optional features of the Document Processing Service.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are defined in [\[MS-OFCGLOS\]](#):

- crawled property**
- crawled property category**
- crawled property set identifier**
- IFilter**
- index schema**
- item processing**
- managed property**
- MIME type**
- phrase break**
- search service application**
- variant type**
- XML namespace prefix**
- XML node**
- XML Path Language (XPath)**
- XML schema definition (XSD)**
- XPath expression**

The following terms are specific to this document:

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-FSCX] Microsoft Corporation, "[Configuration \(XML-RPC\) Protocol Specification](#)".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>

### 1.2.2 Informative References

[MS-FSAS] Microsoft Corporation, "[Administration Services Protocol Specification](#)".

[MS-FSCMW] Microsoft Corporation, "[Configuration Middleware Protocol Specification](#)".

[MS-FSFIXML] Microsoft Corporation, "[FIXML Data Structure](#)".

[MS-FSSCFG] Microsoft Corporation, "[Search Configuration File Format Specification](#)".

[MS-OFGLGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

### 1.3 Structure Overview (Synopsis)

This document specifies a configuration file set that is used by the Document Processing Service. The configuration file set contains a mix of static configuration parameters and configuration parameters derived from the index schema. The index schema is generated by the Administration Service. The static configuration parameters are required in order for the specified services to operate correctly. The services download these configuration files from the Configuration Service.

Part of the configuration file set is derived from the **index schema** and specifies configuration parameters for the **item processing** that occurs prior to indexing. The configuration parameters derived from the index schema represent configuration that is dependent of the actual configuration of the extended **search service application** when it comes to configurable search-related features and **managed property** settings. The configuration files related to managed properties and **crawled properties** are specified in ManagedProperties.xml (section [2.2](#)) and PropertyCategories.xml (section [2.3](#)).

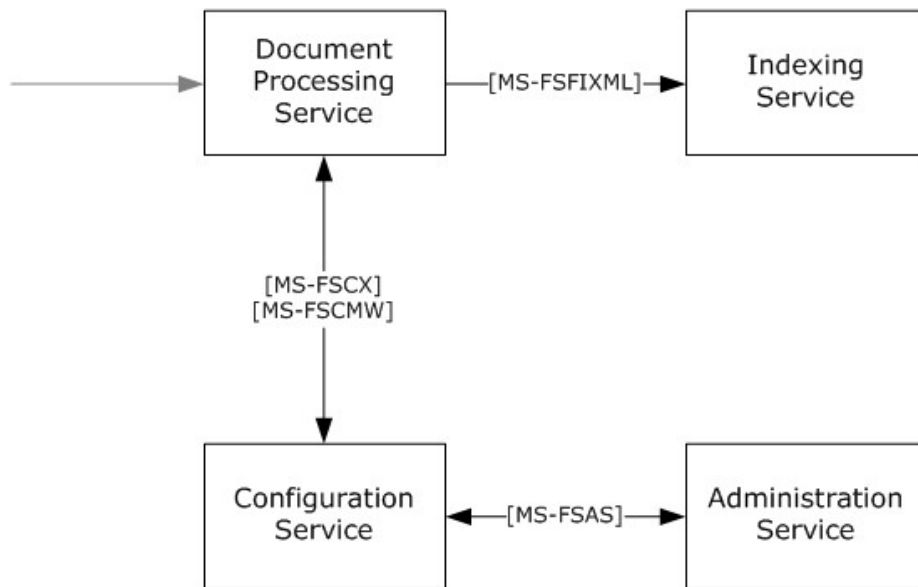
The next file of the configuration file set contains file extensions that bypass the default **MIME type** detection during item processing in the Document Processing Service. This configuration file is specified in user\_converter\_rules.xml (section [2.1](#)).

The next file of the configuration file set contains crawled properties that can be created from specific parts of the crawled XML items. This feature is targeted for web and shared folder crawling of stand-alone XML items that require customized extraction and transformation. This configuration file is specified in XMLMapper.xml (section [2.5](#)).

The last file of the configuration file set specifies whether to enable or disable optional item processing features of the Document Processing Service. This configuration file is specified in section OptionalProcessing.xml (section [2.4](#)).

### 1.4 Relationship to Protocols and Other Structures

The following figure provides a high level overview of the services and protocols for the configuration files specified in this document.



**Figure 1: Overview of services and protocols for the configuration files**

The configuration files described in this document are used by the Document Processing Service. The Document Processing Service also uses the Search Configuration File Format as described in [\[MS-FSSCFG\]](#), which are additional configuration files fully or partly derived from the index schema. The Document Processing Service forwards processed items to the Indexing Service using the format described in [\[MS-FSFXML\]](#).

All specified configuration files are stored on the Configuration Service and transferred by using [\[MS-FSCX\]](#) or [\[MS-FSCMW\]](#). These configuration files may be changed through the Administration Service as described in [\[MS-FSAS\]](#).

### 1.5 Applicability Statement

The file format structures specified in this document are applicable for full-text search applications.

### 1.6 Versioning and Localization

None.

### 1.7 Vendor-Extensible Fields

None.



## 2 Structures

For each configuration file specified in this document, the section begins with a table equal to the table specified. The derived requirements on a protocol implementation are specified in the following table.

Requirement	Details
<a href="#">[MS-FSCX]</a> storage path	The specified file path MUST be used when requesting the file by using the [MS-FSCX] protocol. For more information, see [MS-FSCX].
Type of data	<ul style="list-style-type: none"><li>Configuration information derived from index schema. The configuration information is used by the services specified under 'Service using file' row in the table.</li><li>Non-configurable protocol-related information. The configuration information is used by the services specified under 'Service using file' row in the table.</li><li>Implementation specific configuration information. These parameters are fixed configuration information specific to the services specified in the 'Service using file' row in the table.</li></ul>
Service using file	This row specifies the services that MUST read parameters from this file. This row MUST be the Document Processing Service, the Administration Service, or both.

The file format for all configuration files specified in this document MUST be XML, containing name/value pairs as XML attributes. The XML elements can be grouped in the XML document. This document provides documentation on the XML file in **XML schema definition (XSD)** syntax.

All XML configuration files MUST be formatted in accordance with the XSD specification for the given configuration file.

The schema specified in this document appears for documentation purposes only. A formal reference to the XSD MUST NOT be present in the XML document passed via the [MS-FSCX] interface.

### 2.1 user\_converter\_rules.xml

The derived requirements for a protocol using this file format are specified in the following table.

Requirement	Details
<a href="#">[MS-FSCX]</a> storage path	DocumentProcessor/formatdetector/user_converter_rules.xml
Type of data	Configuration information for format detection
Service using file	Document Processing Service

This file contains file extensions that bypass the default MIME type detection in the Document Processing Service. Any file extension that is not supported by default MUST be added to this file along with its associated MIME type in order for it to be routed to the correct **IFilter**.

#### 2.1.1 Part Enumerations

None.

## 2.1.2 Extensions

None.

## 2.1.3 Global Elements

The **ConverterRules** element MUST contain a single **CT\_IFilter** element, as specified in CT\_IFilter (section [2.1.5.1](#)), and a single **CT\_MimeMapping** element, as specified in CT\_MimeMapping (section [2.1.5.4](#)). This element specifies a mapping between IFilters and MIME types.

The XML MUST adhere to the following XSD.

```
<xs:element name="ConverterRules">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="IFilter" type="CT_IFilter" minOccurs="1"
        maxOccurs="1" />
      <xs:element name="MimeMapping" type="CT_MimeMapping" minOccurs="1"
        maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

## 2.1.4 Global Attributes

None.

## 2.1.5 Complex Types

### 2.1.5.1 CT\_IFilter

The **CT\_IFilter** element MUST contain zero or more **CT\_Trust** elements, as specified in CT\_Trust (section [2.1.5.2](#)). The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_IFilter">
  <xs:sequence>
    <xs:element name="trust" type="CT_Trust" minOccurs="0"
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

### 2.1.5.2 CT\_Trust

The **CT\_Trust** element MUST contain zero or more **CT\_Ext** elements, as specified in CT\_Ext (section [2.1.5.3](#)). The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_Trust">
  <xs:sequence>
    <xs:element name="ext" type="CT_Ext" maxOccurs="unbounded" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

### 2.1.5.3 CT\_Ext

The **CT\_Ext** element specifies a file extension that bypasses the default MIME type detection performed by the Document Processing Service. The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_Ext">
  <xs:attribute name="name" type="ST_Name" use="required" />
  <xs:attribute name="mimetype" type="ST_MimeType" use="required" />
</xs:complexType>
```

### 2.1.5.4 CT\_MimeMapping

The **CT\_MimeMapping** element contains zero or more **CT\_Mime** elements, as specified in CT\_Mime (section [2.1.5.5](#)). The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_MimeMapping">
  <xs:sequence>
    <xs:element name="mime" type="CT_Mime" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

### 2.1.5.5 CT\_Mime

The **CT\_Mime** element specifies a mapping between a MIME type identified by a **CT\_Ext** element, as specified in CT\_Ext (section [2.1.5.3](#)), and a textual description of that MIME type. The textual description of the MIME type is the content of this element. The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_Mime">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="type" type="ST_Mime" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

## 2.1.6 Simple Types

### 2.1.6.1 ST\_Mime

A MIME type that MUST match a single MIME type attribute in the set of **CT\_Ext** attributes, as specified in section CT\_Ext (section [2.1.5.3](#)). The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_Mime">
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>
```

### 2.1.6.2 ST\_MimeType

Specifies the MIME type associated with the file extension. The XML MUST adhere to the following XSD.

```

<xs:simpleType name="ST_MimeType">
  <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>

```

### 2.1.6.3 ST\_Name

Specifies the name associated with the file extension. The XML MUST adhere to the following XSD.

```

<xs:simpleType name="ST_Name">
  <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>

```

## 2.2 ManagedProperties.xml

The derived requirements for a protocol using this file format are specified in the following table.

Requirement	Details
[MS-FSCX] storage path	DocumentProcessor/ManagedProperties.xml
Type of data	Configuration information for managed properties
Service using file	Document Processing Service Administration Service

This file specifies mappings between managed properties and crawled properties.

### 2.2.1 Part Enumerations

None.

### 2.2.2 Extensions

None.

### 2.2.3 Global Elements

The **properties** element controls if the value of the managed property is selected from a single crawled property, or if all values are merged. The XML MUST adhere to the following XSD.

```

<xs:element name="properties">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="select" type="CT_select" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element name="merge" type="CT_merge" minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

## 2.2.4 Global Attributes

None.

## 2.2.5 Complex Types

### 2.2.5.1 CT\_select

This element contains a mapping from a nonempty set of crawled properties to a managed property.

This element MUST contain a nonempty set of **CT\_property** elements, as specified in CT\_property (section [2.2.5.3](#)). When more than one **CT\_property** elements are specified, the first **CT\_property** name matching a given crawled property MUST be used to create the mapping.

The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_select">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="1" type="CT_property"
      name="property" />
  </xs:sequence>
  <xs:attribute name="name" type="ST_name" use="required" />
  <xs:attribute name="type" type="ST_type" use="required" />
</xs:complexType>
```

### 2.2.5.2 CT\_merge

This element contains the mapping from a nonempty set of crawled properties to a single managed property.

This element MUST contain a nonempty set of **CT\_property** elements, as specified in CT\_property (section [2.2.5.3](#)). When more than one **CT\_property** elements are specified, the merged set of all matching crawled properties MUST be used to create the mapping. The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_merge">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="1" type="CT_property"
      name="property" />
  </xs:sequence>
  <xs:attribute name="name" type="ST_name" use="required" />
  <xs:attribute name="type" type="ST_type" use="required" />
</xs:complexType>
```

### 2.2.5.3 CT\_property

This element specifies the name of the crawled property. The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_property">
  <xs:attribute name="name" type="ST_cpname" use="required" />
</xs:complexType>
```

## 2.2.6 Simple Types

### 2.2.6.1 ST\_name

This type specifies the name of the managed property. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_name">
  <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
```

### 2.2.6.2 ST\_cpname

This type specifies the name of the crawled property. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_cpname">
  <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
```

### 2.2.6.3 ST\_type

This type specifies the valid data types for the managed property. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="string"/>
    <xs:enumeration value="boolean"/>
    <xs:enumeration value="int"/>
    <xs:enumeration value="float"/>
    <xs:enumeration value="decimal"/>
    <xs:enumeration value="datetime"/>
  </xs:restriction>
</xs:simpleType>
```

## 2.3 PropertyCategories.xml

The derived requirements for a protocol using this file format are specified in the following table.

Requirement	Details
<a href="#">[MS-FSCX]</a> storage path	DocumentProcessor/PropertyCategories.xml
Type of data	Configuration information for crawled properties
Service using file	Document Processing Service Administration Service

This file contains mappings between **crawled property categories** and **crawled property set identifiers**.

### 2.3.1 Part Enumerations

None.

### 2.3.2 Extensions

None.

### 2.3.3 Global Elements

This element MUST contain a nonempty set of **CT\_category** elements, as specified in CT\_category (section [2.3.5.1](#)), and a nonempty set of **CT\_indexedproperties** elements, as specified in CT\_indexedproperties (section [2.3.5.2](#)). The XML MUST adhere to the following XSD.

```
<xs:element name="properties">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="category" type="CT_category" minOccurs="1"
        maxOccurs="unbounded" />
      <xs:element name="indexed-properties" type="CT_indexedproperties"
        minOccurs="1" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 2.3.4 Global Attributes

None.

### 2.3.5 Complex Types

#### 2.3.5.1 CT\_category

This element contains a mapping between a crawled property category and one or more crawled property set identifiers.

This element MUST contain one or more **CT\_propset** elements, as specified in CT\_propset (section [2.3.5.4](#)). The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_category">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="1" type="CT_propset"
      name="propset" />
  </xs:sequence>
  <xs:attribute name="name" type="ST_name" use="required" />
  <xs:attribute name="indexed" type="ST_indexed" use="required" />
  <xs:attribute name="discover" type="ST_discover" use="required" />
</xs:complexType>
```

#### 2.3.5.2 CT\_indexedproperties

This element contains crawled properties contained in the crawled property set identifiers specified by the **CT\_category** attribute, as specified in CT\_category (section [2.3.5.1](#)). This element MUST

contain zero or more **CT\_property** elements, as specified in CT\_property (section [2.2.5.3](#)). The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_indexedproperties">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" type="CT_property"
      name="property" />
  </xs:sequence>
</xs:complexType>
```

### 2.3.5.3 CT\_property

This element specifies a crawled property name. The crawled property category of the crawled property MUST be specified by a **CT\_category** element, as specified in CT\_category (section [2.3.5.1](#)). If the **ST\_indexed** attribute, as specified in ST\_indexed (section [2.3.6.2](#)), of the **CT\_category** element is "yes", the **ST\_cpindexed** attribute, as specified in ST\_cpindexed (section [2.3.6.4](#)), MUST be "no". If the **ST\_indexed** attribute of the **CT\_category** element is "no", the **ST\_cpindexed** attribute MUST be "yes". The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_property">
  <xs:attribute name="name" type="ST_cpname" use="required" />
  <xs:attribute name="indexed" type="ST_cpindexed" use="required" />
</xs:complexType>
```

### 2.3.5.4 CT\_propset

This element specifies the name of the crawled property set identifier. The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_propset">
  <xs:attribute name="name" type="ST_propset" use="required" />
</xs:complexType>
```

## 2.3.6 Simple Types

### 2.3.6.1 ST\_name

This type specifies the name of the crawled property category. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_name">
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>
```

### 2.3.6.2 ST\_indexed

The meanings of the allowed ST\_indexed values are specified in the following table.



Value	Meaning
yes	This specifies that all crawled properties matching this crawled property category MUST be indexed.
no	This specifies that this crawled property category MUST NOT be indexed.

The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_indexed">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.3.6.3 ST\_cpname

This type specifies the name of the crawled property. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_cpname">
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>
```

### 2.3.6.4 ST\_cpindexed

The meanings of the allowed ST\_cpindexed values are specified in the following table.

Value	Meaning
yes	This specifies that this crawled property MUST be indexed.
no	This specifies that this crawled property MUST NOT be indexed.

The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_cpindexed">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.3.6.5 ST\_discover

The meanings of the allowed ST\_discover values are specified in the following table.

Value	Meaning
yes	Specifies that all crawled properties matching this crawled property category MUST be reported to the Schema Manager in the Administration Service.

Value	Meaning
no	Specifies that this crawled property MUST NOT be reported.

The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_discover">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.3.6.6 ST\_propset

This type specifies the name of the crawled property set identifier. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_propset">
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>
```

## 2.4 OptionalProcessing.xml

The derived requirements for a protocol using this file format are specified in the following table.

Requirement	Details
<a href="#">[MS-FSCX]</a> storage path	DocumentProcessor/OptionalProcessing.xml
Type of data	Implementation-specific configuration for enabling optional item processing features
Service using file	Document Processing Service

This file specifies whether to enable or disable optional item processing features of the Document Processing Service.

### 2.4.1 Part Enumerations

None.

### 2.4.2 Extensions

None.

### 2.4.3 Global Elements

The **optionalprocessing** element specifies a nonempty set of item processing features, where each feature is specified by the **processor** global attribute, as specified in CT\_processor (section [2.4.5.1](#)). The XML MUST adhere to the following XSD.

```

<xs:element name="optionalprocessing">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="unbounded" type="CT_processor"
        name="processor" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

## 2.4.4 Global Attributes

None.

## 2.4.5 Complex Types

### 2.4.5.1 CT\_processor

This element specifies an item processing feature, and whether the feature is enabled or disabled. The XML MUST adhere to the following XSD.

```

<xs:complexType name="CT_processor">
  <xs:attribute name="name" type="ST_name" use="required" />
  <xs:attribute name="active" type="ST_active" use="required" />
</xs:complexType>

```

## 2.4.6 Simple Types

### 2.4.6.1 ST\_name

This type specifies the name of the item processing feature. The XML MUST adhere to the following XSD.

```

<xs:simpleType name="ST_name">
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>

```

### 2.4.6.2 ST\_active

The meanings of the allowed ST\_active values are specified in the following table.

Value	Meaning
yes	This specifies that the item processing feature MUST be enabled
no	This specifies that the item processing feature MUST NOT be enabled

The XML MUST adhere to the following XSD.

```

<xs:simpleType name="ST_active">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
  </xs:restriction>
</xs:simpleType>

```

```

    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>

```

## 2.5 XMLMapper.xml

The derived requirements for a protocol using this file format are specified in the following table.

Requirement	Details
[MS-FSCX] storage path	DocumentProcessor/XMLMapper.xml
Type of data	Implementation-specific configuration of crawled properties that can be created from specific parts of the crawled XML items
Service using file	Document Processing Service

The item processing pipeline in uses the XMLMapper.xml configuration file to create crawled properties from specific parts of crawled XML items. The content to extract from the crawled item is expressed in **XPath** together with functionality provided by the configuration file itself.

Note that a crawled item might consist of multiple parts, for example, a list item or an e-mail with multiple attachments. The item processing pipeline only uses the XMLMapper.xml configuration on the first part.

### 2.5.1 Part Enumerations

None.

### 2.5.2 Extensions

None.

### 2.5.3 Global Elements

The **XMLPropertiesCreator** element MUST contain one default crawled property category with the **propset** element, as specified by ST\_propset in section [2.5.6.1](#), one default **variant type (2)** with the **type** element, as specified by ST\_type in section [2.5.6.23](#), one set of mappings from **XPath expressions** to crawled properties with the **XMLMappings** elements, as specified by CT\_XMLMappings in section [2.5.5.1](#), and an optional **phrase break** for crawled properties with the **paragraph-sep** element, as specified by ST\_paragraph-sep in section [2.5.6.2](#).

The XML MUST adhere to the following XSD.

```

<xs:element name="XMLPropertiesCreator">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="propset" type="ST_propset" minOccurs="1" maxOccurs="1"/>
      <xs:element name="type" type="ST_type" minOccurs="1" maxOccurs="1"/>
      <xs:element name="paragraph-sep" type="ST_paragraph-sep" minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="XMLMappings" minOccurs="1" maxOccurs="1"
        type="CT_XMLMappings" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```
</xs:complexType>
</xs:element>
```

## 2.5.4 Global Attributes

None.

## 2.5.5 Complex Types

### 2.5.5.1 CT\_XMLMappings

This element specifies a set of mappings from XPath expressions to crawled properties. This element contains a set of **Namespace** elements, as specified by CT\_Namespace in section [2.5.5.2](#), a set of **SubTree** elements, as specified by CT\_SubTree in section [2.5.5.3](#), a set of **Mapping** elements, as specified by CT\_Mapping in section [2.5.5.4](#), and a set of **MappingGroup** elements, as specified by CT\_MappingGroup in section [2.5.5.5](#).

The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_XMLMappings">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Namespace"
      type="CT_Namespace" />
    <xs:element name="SubTree" type="CT_SubTree" minOccurs="0"
      maxOccurs="unbounded" />
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Mapping"
      type="CT_Mapping" />
    <xs:element name="MappingGroup" minOccurs="0" maxOccurs="unbounded"
      type="CT_MappingGroup" />
  </xs:sequence>
</xs:complexType>
```

### 2.5.5.2 CT\_Namespace

This element specifies an **XML namespace prefix** for use in XPath expressions, and MUST contain one **name** attribute, as specified by ST\_name in section [2.5.6.3](#), and one **uri** attribute, as specified by ST\_uri in section [2.5.6.4](#).

The XML MUST adhere to the following XSD.

```
<xs:complexType name="CT_Namespace">
  <xs:attribute name="name" type="ST_name" use="required" />
  <xs:attribute name="uri" type="ST_uri" use="required" />
</xs:complexType>
```

### 2.5.5.3 CT\_SubTree

This element specifies a common XPath base path for the contained set of **Mapping** elements (section [2.5.5.4](#)), **MappingGroup** elements (section [2.5.5.5](#)), and **SubTree** elements (section [2.5.5.3](#)). This element MUST contain a **base-path** attribute, as specified by ST\_base-path in section [2.5.6.16](#).

The XML MUST adhere to the following XSD.

```

<xs:complexType name="CT_SubTree">
  <xs:sequence>
    <xs:element name="SubTree" type="CT_SubTree"
      minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="Mapping" type="CT_Mapping"
      minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="MappingGroup" type="CT_MappingGroup"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="base-path" type="ST_base-path" use="required" />
</xs:complexType>

```

#### 2.5.5.4 CT\_Mapping

This element specifies a mapping from an XPath expression to a crawled property. This element **MUST** specify a **path** attribute, as specified by ST\_path in section [2.5.6.5](#), and optionally the **attr** attribute (section [2.5.6.6](#)), the **type** attribute (section [2.5.6.8](#)), the **propset** attribute (section [2.5.6.7](#)), the **sep-str** attribute (section [2.5.6.9](#)), the **pre-str** attribute (section [2.5.6.10](#)), the **post-str** attribute (section [2.5.6.11](#)), the **ignore-whitespace** attribute (section [2.5.6.12](#)), the **strip-tags** attribute (section [2.5.6.13](#)), the **shallow** attribute (section [2.5.6.14](#)), and the **mode** attribute (section [2.5.6.15](#)).

The XML **MUST** adhere to the following XSD.

```

<xs:complexType name="CT_Mapping">
  <xs:attribute name="path" type="ST_path" use="required" />
  <xs:attribute name="attr" type="ST_attr" use="optional" />
  <xs:attribute name="type" type="ST_m-type" use="optional" />
  <xs:attribute name="propset" type="ST_m-propset" use="optional" />
  <xs:attribute name="sep-str" type="ST_sep-str" use="optional" />
  <xs:attribute name="pre-str" type="ST_pre-str" use="optional" />
  <xs:attribute name="post-str" type="ST_post-str" use="optional" />
  <xs:attribute name="ignore-whitespace"
    type="ST_ignore-whitespace" use="optional" />
  <xs:attribute name="strip-tags" type="ST_strip-tags" use="optional" />
  <xs:attribute name="shallow" type="ST_shallow" use="optional" />
  <xs:attribute name="mode" type="ST_mode" use="optional" />
</xs:complexType>

```

#### 2.5.5.5 CT\_MappingGroup

This element specifies a mapping between the crawled contents of the containing **Mapping** elements (specified in section [2.5.5.4](#)) to a crawled property. This element **MUST** specify an **attr** attribute (section [2.5.6.6](#)) and a **base-path** attribute (section [2.5.6.17](#)), and optionally a **propset** attribute (section [2.5.6.7](#)), a **type** attribute (section [2.5.6.8](#)), a **mode** attribute (section [2.5.6.22](#)), a **sep-str** attribute (section [2.5.6.9](#)), a **pre-str** attribute (section [2.5.6.10](#)), a **post-str** attribute (section [2.5.6.11](#)), a **rec-sep-str** attribute (section [2.5.6.18](#)), a **rec-pre-str** attribute (section [2.5.6.19](#)), a **rec-post-str** attribute (section [2.5.6.20](#)), and a **select** attribute (section [2.5.6.21](#)).

The XML **MUST** adhere to the following XSD.

```

<xs:complexType name="CT_MappingGroup">
  <xs:sequence>
    <xs:element name="Mapping" type="CT_Mapping" minOccurs="0"

```

```

        maxOccurs="unbounded" />
</xs:sequence>
<xs:attribute name="attr" type="ST_attr" use="required" />
<xs:attribute name="base-path" type="ST_mg-base-path" use="required" />
<xs:attribute name="propset" type="ST_m-propset" use="optional" />
<xs:attribute name="type" type="ST_m-type" use="optional" />
<xs:attribute name="mode" type="ST_mg-mode" use="optional" />
<xs:attribute name="sep-str" type="ST_sep-str" use="optional" />
<xs:attribute name="pre-str" type="ST_pre-str" use="optional" />
<xs:attribute name="post-str" type="ST_post-str" use="optional" />
<xs:attribute name="rec-sep-str" type="ST_mg-rec-sep-str" use="optional" />
<xs:attribute name="rec-pre-str" type="ST_mg-rec-pre-str" use="optional" />
<xs:attribute name="rec-post-str" type="ST_mg-rec-post-str"
        use="optional" />
<xs:attribute name="select" type="ST_mg-select"
        use="optional" />
</xs:complexType>

```

## 2.5.6 Simple Types

### 2.5.6.1 ST\_propset

This type specifies the default crawled property category for the crawled properties. The default property category can be overridden by the individual **Mapping** or **MappingGroup** elements (specified in section [2.5.5.4](#) and section [2.5.5.5](#), respectively), using the **propset** attribute, as specified in section [2.5.6.7](#). The XML MUST adhere to the following XSD.

```

<xs:simpleType name="ST_propset">
  <xs:restriction base="xs:string" />
</xs:simpleType>

```

### 2.5.6.2 ST\_paragraph-sep

This type specifies the phrase break for the created crawled properties. The XML MUST adhere to the following XSD.

```

<xs:simpleType name="ST_paragraph-sep">
  <xs:restriction base="xs:string" />
</xs:simpleType>

```

### 2.5.6.3 ST\_name

This type specifies the XML namespace prefix to use in XPath expressions. The XML MUST adhere to the following XSD.

```

<xs:simpleType name="ST_name">
  <xs:restriction base="xs:string" />
</xs:simpleType>

```

#### 2.5.6.4 ST\_uri

This type specifies the XML namespace prefix to use in XPath expressions. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_uri">  
  <xs:restriction base="xs:string" />  
</xs:simpleType>
```

#### 2.5.6.5 ST\_path

This type specifies the XPath expression to the **XML node** to extract content from when creating the crawled property. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_path">  
  <xs:restriction base="xs:string" />  
</xs:simpleType>
```

#### 2.5.6.6 ST\_attr

This type specifies the name of the created crawled property. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_attr">  
  <xs:restriction base="xs:string" />  
</xs:simpleType>
```

#### 2.5.6.7 ST\_m-propset

This type overrides the default crawled property category for the crawled properties. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_m-propset">  
  <xs:restriction base="xs:string" />  
</xs:simpleType>
```

#### 2.5.6.8 ST\_m-type

This type overrides the default variant type for the crawled property. The meaning of the variant type is specified in the following mapping table.

Variant Type	Meaning
3	Int
4	Float
11	Bool
14	Decimal
31	String



Variant Type	Meaning
64	Datetime

The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_m-type">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="65535" />
  </xs:restriction>
</xs:simpleType>
```

### 2.5.6.9 ST\_sep-str

This type specifies the separator string when the **path**, as specified in section [2.5.6.5](#), matches more than one XML node. The default value for this type MUST be the " " (space) character. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_sep-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>
```

### 2.5.6.10 ST\_pre-str

This type specifies a string that MUST be used as the prefix of the extracted content. The default value for this type MUST be an empty string. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_pre-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>
```

### 2.5.6.11 ST\_post-str

This type specifies a string that MUST be used as a postfix of the extracted content. The default value for this type MUST be an empty string. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_post-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>
```

### 2.5.6.12 ST\_ignore-whitespace

This type specifies whether sequences of whitespace characters in the extracted content can be replaced with a single instance of the whitespace character or not. The meanings of the allowed ST\_ignore-whitespace values are specified in the following table.

Value	Meaning
yes	Replace sequence with single instance of whitespace

Value	Meaning
no	Do not replace sequences of whitespace with single instance

The default value MUST be "yes". The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_ignore-whitespace">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.5.6.13 ST\_strip-tags

This type specifies whether XML tags can be removed from the extracted content or not. The meanings of the allowed ST\_strip-tags values are specified in the following table.

Value	Meaning
yes	Remove tags
no	Do not remove tags

The default value MUST be "yes". The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_strip-tags">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.5.6.14 ST\_shallow

This type specifies the whether to extract only child XML nodes or if extraction can be recursive for child XML nodes. The meanings of the allowed ST\_shallow values are specified in the following table.

Value	Meaning
yes	Only extract XML child nodes
no	Extract XML child nodes recursively

The default value MUST be "no". The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_shallow">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>
```

```
</xs:simpleType>
```

### 2.5.6.15 ST\_mode

This type how to create the crawled property when a different **Mapping** or **MappingGroup** element (specified in section [2.5.5.4](#) and section [2.5.5.5](#), respectively), has already extracted content to the same crawled property. The meanings of the allowed ST\_mode values are specified in the following table.

Value	Meaning
append	Append content to existing property
prepend	Prepend content to existing property
overwrite	Overwrite content of existing property

The default value MUST be "append". The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_mode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="append"/>
    <xs:enumeration value="prepend"/>
    <xs:enumeration value="overwrite"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.5.6.16 ST\_base-path

This type specifies the XPath base path for the **Mapping** and **MappingGroup** elements (specified in section [2.5.5.4](#) and section [2.5.5.5](#), respectively), that are specified within the **SubTree** element (section [2.5.5.3](#)). The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_base-path">
  <xs:restriction base="xs:string" />
</xs:simpleType>
```

### 2.5.6.17 ST\_mg-base-path

This type specifies the XPath base path for the **Mapping** and **MappingGroup** elements (specified in section [2.5.5.4](#) and section [2.5.5.5](#), respectively). If the **Mapping** or **MappingGroup** are specified within a **SubTree** element, as specified in section [2.5.5.3](#), the base path specified by the **SubTree** element MUST be overridden with this base path. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_mg-base-path">
  <xs:restriction base="xs:string" />
</xs:simpleType>
```

### 2.5.6.18 ST\_mg-rec-sep-str

This type specifies the string separator used between extracted content from each **Mapping** element (section [2.5.5.4](#)) specified in this group. The default value MUST be the " " (space) character. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_mg-rec-sep-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>
```

### 2.5.6.19 ST\_mg-rec-pre-str

This type specifies a string to prepend to the first **Mapping** element (section [2.5.5.4](#)) in this group. The default value MUST be the empty string. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_mg-rec-pre-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>
```

### 2.5.6.20 ST\_mg-rec-post-str

This type specifies a string to append to the last **Mapping** element (section [2.5.5.4](#)) in this group. The default value MUST be the empty string. The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_mg-rec-post-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>
```

### 2.5.6.21 ST\_mg-select

This type specifies which **Mapping** elements (section [2.5.5.4](#)) to extract content from.

Value	Meaning
merge	Merge content from all elements using the <b>rec-sep-str</b> , <b>rec-pre-str</b> , and <b>rec-post-str</b> attributes.
first	Extract content from the first element that has content
longest	Extract content from the element with the longest content

The default value MUST be "merge". The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_mg-select">
  <xs:restriction base="xs:string">
    <xs:enumeration value="merge"/>
    <xs:enumeration value="first"/>
    <xs:enumeration value="longest"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.5.6.22 ST\_mg-mode

This type how to create the crawled property when a different **Mapping** or **MappingGroup** element (specified in section [2.5.5.4](#) and section [2.5.5.5](#), respectively), has already extracted content to the same crawled property. The meanings of the allowed ST\_mg-mode values are specified in the following table.

Value	Meaning
append	Undefined, prepend or append with existing property will occur
prepend	Undefined, prepend or append with existing property will occur
overwrite	Overwrite content of existing property

The default value MUST be "append". The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_mg-mode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="append"/>
    <xs:enumeration value="prepend"/>
    <xs:enumeration value="overwrite"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.5.6.23 ST\_type

This type specifies the default variant type for the crawled property. The meaning of the variant type is specified in the following mapping table.

Variant Type	Meaning
3	Int
4	Float
11	Bool
14	Decimal
31	String
64	Datetime

The default variant type can be overridden by **Mapping** or **MappingGroup** elements (specified in section [2.5.5.4](#) and section [2.5.5.5](#), respectively), using the **type** attribute, as specified in section [2.5.6.8](#).

The XML MUST adhere to the following XSD.

```
<xs:simpleType name="ST_type">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="65535" />
  </xs:restriction>
```

```
</xs:simpleType>
```

## 2.6 CustomPropertyExtractors.xml

The derived requirements for a protocol using this file format are specified in the following table.

Requirement	Details
[MS-FSCX] storage path	DocumentProcessor/ CustomPropertyExtractors.xml
Type of data	Implementation-specific configuration for enabling optional item processing features for property extraction.
Service using file	Document Processing Service

This file specifies the setup of the optional property extractors of the Document Processing Service.

### 2.6.1 Part Enumerations

None.

### 2.6.2 Extensions

None.

### 2.6.3 Global Elements

The **extractors** element MUST contain one set of mappings from **extractor** elements, as specified in section [2.6.5.1](#). The XML MUST adhere to the following XML schema.

```
<xs:element name="extractors">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="extractor" minOccurs="1" maxOccurs="unbounded" type="CT_extractor"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 2.6.4 Global Attributes

None.

### 2.6.5 Complex Types

#### 2.6.5.1 CT\_extractor

The **CT\_extractor** element specifies one property extractor. This element MUST contain one **dictionary** element as specified in section [2.6.5.2](#). This element MUST contain a **name** attribute as specified in section [2.6.6.1](#), a **type** attribute as specified in section [2.6.6.2](#), and a **property** attribute as specified in section [2.6.6.3](#). The XML MUST adhere to the following XML schema.

```
<xs:complexType name="CT_extractor">
```

```

<xs:sequence>
  <xs:element name="dictionary" minOccurs="1" maxOccurs="1" type="CT_dictionary"/>
</xs:sequence>
<xs:attribute name="name" type="ST_name" use="required"/>
<xs:attribute name="type" type="ST_type" use="required"/>
<xs:attribute name="property" type="ST_property" use="required"/>
</xs:complexType>

```

### 2.6.5.2 CT\_dictionary

The **CT\_dictionary** element specifies the dictionary and how to use its values in the extractor. This element **MUST** contain one **name** attribute as specified in section [2.6.6.4](#), and it can contain a **yield-values** attribute as specified in section [2.6.6.5](#). The XML **MUST** adhere to the following XML schema.

```

<xs:complexType name="CT_dictionary">
  <xs:attribute name="name" type="ST_dictionary_name" use="required"/>
  <xs:attribute name="yield-values" type="ST_yield_values" use="optional" default="no"/>
</xs:complexType>

```

## 2.6.6 Simple Types

### 2.6.6.1 ST\_name

The **ST\_name** type specifies the name of the extractor. The XML **MUST** adhere to the following XML schema.

```

<xs:simpleType name="ST_name">
  <xs:restriction base="xs:string" />
</xs:simpleType>

```

### 2.6.6.2 ST\_type

The **ST\_type** type specifies the type of the extractor, as described in the following table.

Variant type	Meaning
"WholeWords"	The extractor performs case-sensitive matching of the dictionary entries in the token stream.
"WordParts"	The extractor performs case-sensitive matching of the dictionary entries in untokenized text.
"Verbatim"	The extractor performs case-insensitive matching of the dictionary entries in the token stream.
"Substring"	The extractor performs case-insensitive matching of the dictionary entries in untokenized text.

The XML **MUST** adhere to the following XML schema.

```

<xs:simpleType name="ST_type">
  <xs:restriction base="xs:string">

```

```

    <xs:enumeration value="WholeWords"/>
    <xs:enumeration value="WordParts"/>
    <xs:enumeration value="Verbatim"/>
    <xs:enumeration value="Substring"/>
  </xs:restriction>
</xs:simpleType>

```

### 2.6.6.3 ST\_property

The **ST\_property** type specifies the property name that the extractor creates. The XML MUST adhere to the following XML schema.

```

<xs:simpleType name="ST_property">
  <xs:restriction base="xs:string" />
</xs:simpleType>

```

### 2.6.6.4 ST\_dictionary\_name

The **ST\_dictionary\_name** type specifies the dictionary name. The XML MUST adhere to the following XML schema.

```

<xs:simpleType name="ST_dictionary_name">
  <xs:restriction base="xs:string" />
</xs:simpleType>

```

### 2.6.6.5 ST\_yield\_values

The **ST\_yield\_values** type specifies whether dictionary values are returned in the extracted property. The meanings of the allowed **ST\_yield\_values** values are specified in the following table.

Value	Meaning
"yes"	Return the value of a matching dictionary entry.
"no"	Ignore the values of matching entries and instead return the matching key.

The default value MUST be "no". The XML MUST adhere to the following XML schema.

```

<xs:simpleType name="ST_yield_values">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>

```



## 3 Structure Examples

### 3.1 ManagedProperties.xml

This example shows how various crawled properties are mapped to the managed properties "title", "description", "size", "author", "created", and "email". Because the element of type **CT\_select** is used, the first matching crawled property is selected for the mapping. The following specifies the example managed property mapping.

```
<?xml version="1.0" encoding="utf-8"?>
<properties>
  <select name="title" type="string">
    <property name="F29F85E0-4FF9-1068-AB91-08002B27B3D9:#2:31" />
    <property name="F29F85E0-4FF9-1068-AB91-08002B27B3D9:#2:30" />
  </select>
  <select name="description" type="string">
    <property name="F29F85E0-4FF9-1068-AB91-08002B27B3D9:#6:31" />
    <property name="F29F85E0-4FF9-1068-AB91-08002B27B3D9:#6:30" />
  </select>
  <select name="size" type="int">
    <property name="b725f130-47ef-101a-a5f1-02608c9eebac:#12:19" />
    <property name="00130329-0000-0130-c000-000000131346:ows_ImageSize:20" />
  </select>
  <select name="author" type="string">
    <property name="F29F85E0-4FF9-1068-AB91-08002B27B3D9:#4:31" />
    <property name="F29F85E0-4FF9-1068-AB91-08002B27B3D9:#4:30" />
    <property name="0F451D82-37A2-4831-A5D7-E5D57C3AD793:primary_author:31" />
    <property name="aa568eec-e0e5-11cf-8fda-00aa00a14f93:6:31" />
  </select>
  <select name="created" type="datetime">
    <property name="f29f85e0-4ff9-1068-ab91-08002b27b3d9:#12:64" />
    <property name="b725f130-47ef-101a-a5f1-02608c9eebac:#15:64" />
  </select>
  <select name="email" type="string">
    <property name="00130329-0000-0130-c000-000000131346:ows_Email:31" />
    <property name="00130329-0000-0130-c000-000000131346:ows_Email_x0020_:31" />
  </select>
</properties>
```

### 3.2 PropertyCategories.xml

This example shows how various crawled property set identifiers are mapped to the crawled property categories "email" and "office". Because the "office" category specifies that the indexed attribute of the **CT\_category** element is "no", new crawled properties with property set identifiers belonging to this category are not indexed. The "email" category specifies that all new crawled properties for this category are indexed except the crawled property "00130329-0000-0130-c000-000000131346:ows\_Email\_x0020\_:31", because this crawled property is listed with indexing disabled in the **CT\_property** element. The following specifies the example crawled property category mapping.

```
<?xml version="1.0" encoding="utf-8"?>
<properties>
  <category name="email" indexed="yes" discover="yes">
    <propset name="00130329-0000-0130-c000-000000131346" />
  </category>
</properties>
```

```

</category>
<category name="office" indexed="no" discover="yes">
  <propset name="f29f85e0-4ff9-1068-ab91-08002b27b3d9" />
</category>
<indexed-properties>
  <property name="00130329-0000-0130-c000-000000131346:ows_EMail_x0020_:31"
    indexed="no" />
</indexed-properties>
</properties>

```

### 3.3 user\_converter\_rules.xml

This example states that files with the .doc file extension bypass the default format detection, and that the .doc file extension is associated with the MIME type "application/msword". The following specifies the example format detection rule configuration.

```

<?xml version="1.0" encoding="utf-8"?>
<ConverterRules>
  <IFilter>
    <trust>
      <ext name=".doc" mimetype="application/msword" />
    </trust>
  </IFilter>

  <MimeMapping>
    <mime type="application/msword">Microsoft Office Word Document</mime>
  </MimeMapping>
</ConverterRules>

```

### 3.4 OptionalProcessing.xml

This example states that the feature named "XMLMapper" and the feature named "FFDDumper" are enabled, and that all other features are disabled. The following specifies the example item processing feature configuration.

```

<?xml version="1.0" encoding="utf-8"?>
<optionalprocessing>
  <processor name="personnameextraction" active="no"/>
  <processor name="XMLMapper" active="yes"/>
  <processor name="OffensiveContentFilter" active="no"/>
  <processor name="FFDDumper" active="yes"/>
  <processor name="WholeWordsExtractor1" active="no"/>
  <processor name="WholeWordsExtractor2" active="no"/>
  <processor name="WholeWordsExtractor3" active="no"/>
  <processor name="WordsPartExtractor1" active="no"/>
  <processor name="WordsPartExtractor2" active="no"/>
</optionalprocessing>

```

### 3.5 XMLMapper.xml

This example first states that the default crawled property set identifier is d6ee4933-09c4-46e3-a5e4-b3787cb4a090 with the **propset** element, and that the default variant type is 31 (string) with the **type** element.

Two **Mapping** elements are then specified. The first element specifies a mapping from the XPath node "//Title" to the crawled property "mytitle", and the second element a mapping from the XPath node "//Size" to the crawled property "mysize". The second element also specifies that the crawled content will be converted to the variant type 3 (int).

Then a **MappingGroup** element is specified with the crawled attribute set to "mymulti", containing two **Mapping** elements, and an XPath base path "//Document". The **select** attribute of the **MappingGroup** specifies that the "mymulti" crawled property will contain the first crawled item matching one of the XPath paths in of the specified **Mapping** elements. The following specifies the example XMLMapper configuration.

```
<?xml version="1.0" encoding="utf-8"?>
<XMLPropertiesCreator>
  <propset>d6ee4933-09c4-46e3-a5e4-b3787cb4a090</propset>
  <type>31</type>
  <XMLMappings>
    <Mapping attr="mytitle" path="//Title"/>
    <Mapping attr="mysize" path="//Size" type="3"/>
    <MappingGroup attr="mymulti" base-path="//Document" select="first">
      <Mapping path="Tutti"/>
      <Mapping path="Frutti"/>
    </MappingGroup>
  </XMLMappings>
</XMLPropertiesCreator>
```

### 3.6 CustomPropertyExtractors.xml

This example shows how to enable a custom property extractor, named "myextractor" of type "Verbatim", that creates a property named "myproperty" and is based on the dictionary named "mydictionary". This extractor is further configured to return the values in its dictionary as matches. The following XML specifies the example feature configuration for item processing.

```
<?xml version="1.0" encoding="utf-8"?>
<extractors>
  <extractor name="myextractor" type="Verbatim" property="myproperty">
    <dictionary name="mydictionary" yield-values="yes" />
  </extractor>
</extractors>
```

## 4 Security Considerations

None.

## 5 Appendix A: Full XML Schemas

### 5.1 user\_converter\_rules.xsd

The following is the full XML schema for this file format specification.

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

  <xs:element name="ConverterRules">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="IFilter" type="CT_IFilter" minOccurs="1"
          maxOccurs="1" />
        <xs:element name="MimeMapping" type="CT_MimeMapping" minOccurs="1"
          maxOccurs="1" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="CT_IFilter">
    <xs:sequence>
      <xs:element name="trust" type="CT_Trust" minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CT_Trust">
    <xs:sequence>
      <xs:element name="ext" type="CT_Ext" maxOccurs="unbounded"
        minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CT_Ext">
    <xs:attribute name="name" type="ST_Name" use="required" />
    <xs:attribute name="mimetype" type="ST_MimeType" use="required" />
  </xs:complexType>
  <xs:complexType name="CT_MimeMapping">
    <xs:sequence>
      <xs:element name="mime" type="CT_Mime" minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CT_Mime">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type" type="ST_Mime" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="ST_Mime">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ST_MimeType">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ST_Name">
```

```

    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

## 5.2 ManagedProperties.xsd

The following is the full XML schema for this file format specification.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="properties">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="select" type="CT_select" minOccurs="0"
          maxOccurs="unbounded" />
        <xs:element name="merge" type="CT_merge" minOccurs="0"
          maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="CT_select">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="1" type="CT_property"
        name="property" />
    </xs:sequence>
    <xs:attribute name="name" type="ST_name" use="required" />
    <xs:attribute name="type" type="ST_type" use="required" />
  </xs:complexType>
  <xs:complexType name="CT_merge">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="1" type="CT_property"
        name="property" />
    </xs:sequence>
    <xs:attribute name="name" type="ST_name" use="required" />
    <xs:attribute name="type" type="ST_type" use="required" />
  </xs:complexType>
  <xs:complexType name="CT_property">
    <xs:attribute name="name" type="ST_cpname" use="required"/>
  </xs:complexType>
  <xs:simpleType name="ST_name">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ST_cpname">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ST_type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="string"/>
      <xs:enumeration value="boolean"/>
      <xs:enumeration value="int"/>
      <xs:enumeration value="float"/>
      <xs:enumeration value="decimal"/>
      <xs:enumeration value="datetime"/>
    </xs:restriction>
  </xs:simpleType>

```

```
</xs:simpleType>
</xs:schema>
```

### 5.3 PropertyCategories.xsd

The following is the full XML schema for this file format specification.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="properties">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="category" type="CT_category" minOccurs="1"
          maxOccurs="unbounded" />
        <xs:element name="indexed-properties" type="CT_indexedproperties"
          minOccurs="1" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="CT_category">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="1" type="CT_propset"
        name="propset" />
    </xs:sequence>
    <xs:attribute name="name" type="ST_name" use="required" />
    <xs:attribute name="indexed" type="ST_indexed" use="required" />
    <xs:attribute name="discover" type="ST_discover" use="required" />
  </xs:complexType>

  <xs:complexType name="CT_propset">
    <xs:attribute name="name" type="ST_propset" use="required"/>
  </xs:complexType>

  <xs:complexType name="CT_indexedproperties">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0" type="CT_property"
        name="property" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_property">
    <xs:attribute name="name" type="ST_cpname" use="required" />
    <xs:attribute name="indexed" type="ST_cpindexed" use="required" />
  </xs:complexType>

  <xs:simpleType name="ST_cpname">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_cpindexed">
    <xs:restriction base="xs:string">
      <xs:enumeration value="yes"/>
      <xs:enumeration value="no"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_name">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
```

```

<xs:simpleType name="ST_indexed">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ST_discover">
  <xs:restriction base="xs:string">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ST_propset">
  <xs:restriction base="xs:string">
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

## 5.4 OptionalProcessing.xsd

The following is the full XML schema for this file format specification.

```

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

  <xs:element name="optionalprocessing">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded"
          type="CT_processor" name="processor" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="CT_processor">
    <xs:attribute name="name" type="ST_name" use="required" />
    <xs:attribute name="active" type="ST_active" use="required" />
  </xs:complexType>
  <xs:simpleType name="ST_name">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ST_active">
    <xs:restriction base="xs:string">
      <xs:enumeration value="yes"/>
      <xs:enumeration value="no"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>

```

## 5.5 XMLMapper.xsd

The following is the full XML schema for this file format specification.



```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="XMLPropertiesCreator">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="propset" type="ST_propset" minOccurs="1"
          maxOccurs="1"/>
        <xs:element name="type" type="ST_type" minOccurs="1" maxOccurs="1"/>
        <xs:element name="paragraph-sep" type="ST_paragraph-sep" minOccurs="0"
          maxOccurs="1"/>
        <xs:element name="XMLMappings" minOccurs="1" maxOccurs="1"
          type="CT_XMLMappings" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="CT_XMLMappings">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Namespace"
        type="CT_Namespace" />
      <xs:element name="SubTree" minOccurs="0" maxOccurs="unbounded"
        type="CT_SubTree" />
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Mapping"
        type="CT_Mapping" />
      <xs:element name="MappingGroup" minOccurs="0" maxOccurs="unbounded"
        type="CT_MappingGroup" />
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="ST_type">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="65535" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_propset">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_paragraph-sep">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_name">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_uri">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_path">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_attr">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_m-propset">

```

```

    <xs:restriction base="xs:string" />
  </xs:simpleType>
  <xs:simpleType name="ST_m-type">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="65535" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ST_sep-str">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_pre-str">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_post-str">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ST_ignore-whitespace">
    <xs:restriction base="xs:string">
      <xs:enumeration value="yes"/>
      <xs:enumeration value="no"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_strip-tags">
    <xs:restriction base="xs:string">
      <xs:enumeration value="yes"/>
      <xs:enumeration value="no"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_shallow">
    <xs:restriction base="xs:string">
      <xs:enumeration value="yes"/>
      <xs:enumeration value="no"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_mode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="append"/>
      <xs:enumeration value="prepend"/>
      <xs:enumeration value="overwrite"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_mg-mode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="append"/>
      <xs:enumeration value="prepend"/>
      <xs:enumeration value="overwrite"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_base-path">
    <xs:restriction base="xs:string" />

```

```

</xs:simpleType>

<xs:simpleType name="ST_mg-base-path">
  <xs:restriction base="xs:string" />
</xs:simpleType>

<xs:simpleType name="ST_mg-rec-sep-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>

<xs:simpleType name="ST_mg-rec-pre-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>

<xs:simpleType name="ST_mg-rec-post-str">
  <xs:restriction base="xs:string" />
</xs:simpleType>
<xs:simpleType name="ST_mg-select">
  <xs:restriction base="xs:string">
    <xs:enumeration value="merge"/>
    <xs:enumeration value="first"/>
    <xs:enumeration value="longest"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="CT_Namespace">
  <xs:attribute name="name" type="ST_name" use="required" />
  <xs:attribute name="uri" type="ST_uri" use="required" />
</xs:complexType>
<xs:complexType name="CT_SubTree">
  <xs:sequence>
    <xs:element name="SubTree" type="CT_SubTree"
      minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="Mapping" type="CT_Mapping"
      minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="MappingGroup" type="CT_MappingGroup"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="base-path" type="ST_base-path" use="required" />
</xs:complexType>
<xs:complexType name="CT_Mapping">
  <xs:attribute name="path" type="ST_path" use="required" />
  <xs:attribute name="attr" type="ST_attr" use="optional" />
  <xs:attribute name="type" type="ST_m-type" use="optional" />
  <xs:attribute name="propset" type="ST_m-propset" use="optional" />
  <xs:attribute name="sep-str" type="ST_sep-str" use="optional" />
  <xs:attribute name="pre-str" type="ST_pre-str" use="optional" />
  <xs:attribute name="post-str" type="ST_post-str" use="optional" />
  <xs:attribute name="ignore-whitespace" use="optional"
    type="ST_ignore-whitespace" />
  <xs:attribute name="strip-tags" type="ST_strip-tags" use="optional" />
  <xs:attribute name="shallow" type="ST_shallow" use="optional" />
  <xs:attribute name="mode" type="ST_mode" use="optional" />
</xs:complexType>

<xs:complexType name="CT_MappingGroup">
  <xs:sequence>
    <xs:element name="Mapping" type="CT_Mapping" minOccurs="0"
      maxOccurs="unbounded" />
  </xs:sequence>

```

```

<xs:attribute name="attr" type="ST_attr" use="required" />
<xs:attribute name="base-path" type="ST_mg-base-path" use="required" />
<xs:attribute name="propset" type="ST_m-propset" use="optional" />
<xs:attribute name="type" type="ST_m-type" use="optional" />
<xs:attribute name="mode" type="ST_mg-mode" use="optional" />
<xs:attribute name="sep-str" type="ST_sep-str" use="optional" />
<xs:attribute name="pre-str" type="ST_pre-str" use="optional" />
<xs:attribute name="post-str" type="ST_post-str" use="optional" />
<xs:attribute name="rec-sep-str" type="ST_mg-rec-sep-str"
use="optional" />
<xs:attribute name="rec-pre-str" type="ST_mg-rec-pre-str"
use="optional" />
<xs:attribute name="rec-post-str" type="ST_mg-rec-post-str"
use="optional" />
<xs:attribute name="select" type="ST_mg-select" use="optional" />
</xs:complexType>

</xs:schema>

```

## 5.6 CustomPropertyExtractors.xsd

The following is the full XML schema for this file format specification.

```

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

  <xs:element name="extractors" type="CT_extractors"/>

  <xs:complexType name="CT_extractors">
    <xs:sequence>
      <xs:element name="extractor" minOccurs="1" maxOccurs="unbounded" type="CT_extractor"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_extractor">
    <xs:sequence>
      <xs:element name="dictionary" minOccurs="1" maxOccurs="1" type="CT_dictionary"/>
    </xs:sequence>
    <xs:attribute name="name" type="ST_name" use="required"/>
    <xs:attribute name="type" type="ST_type" use="required"/>
    <xs:attribute name="property" type="ST_property" use="required"/>
  </xs:complexType>

  <xs:complexType name="CT_dictionary">
    <xs:attribute name="name" type="ST_dictionary_name" use="required"/>
    <xs:attribute name="yield-values" type="ST_yield_values" use="optional" default="no"/>
  </xs:complexType>

  <xs:simpleType name="ST_name">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="WholeWords"/>
      <xs:enumeration value="WordParts"/>
    </xs:restriction>
  </xs:simpleType>

```

```
        <xs:enumeration value="Verbatim"/>
        <xs:enumeration value="Substring"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ST_property">
    <xs:restriction base="xs:string">
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ST_dictionary_name">
    <xs:restriction base="xs:string">
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="ST_yield_values">
    <xs:restriction base="xs:string">
        <xs:enumeration value="yes"/>
        <xs:enumeration value="no"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>
```

## 6 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® FAST™ Search Server 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

## 7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

## 8 Index

### A

[Applicability](#) 8  
Attributes - global  
  [CustomPropertyExtractors.xml](#) 30  
  [ManagedProperties.xml](#) 13  
  [OptionalProcessing.xml](#) 19  
  [PropertyCategories.xml](#) 15  
  [user\\_converter\\_rules.xml](#) 10  
  [XMLMapper.xml](#) 21

### C

[Change tracking](#) 47  
Common data types and fields ([section 2 9](#), [section 2 9](#))  
Complex types  
  [CT\\_dictionary](#) 31  
  [CT\\_extractor](#) 30  
  [CT\\_Mapping](#) 22  
  [CT\\_MappingGroup](#) 22  
  [CT\\_Namespace](#) 21  
  [CT\\_SubTree](#) 21  
  [CT\\_XMLMappings](#) 21  
  [CT\\_category](#) 15  
  [CT\\_Ext](#) 11  
  [CT\\_IFilter](#) 10  
  [CT\\_indexedproperties](#) 15  
  [CT\\_merge](#) 13  
  [CT\\_Mime](#) 11  
  [CT\\_MimeMapping](#) 11  
  [CT\\_processor](#) 19  
  CT\_property ([section 2.2.5.3](#) 13, [section 2.3.5.3](#) 16)  
  [CT\\_propset](#) 16  
  [CT\\_select](#) 13  
  [CT\\_Trust](#) 10  
[CT\\_dictionary complex type](#) 31  
[CT\\_extractor complex type](#) 30  
[CT\\_Mapping complex type](#) 22  
[CT\\_MappingGroup complex type](#) 22  
[CT\\_Namespace complex type](#) 21  
[CT\\_SubTree complex type](#) 21  
[CT\\_XMLMappings complex type](#) 21  
[CT\\_category complex type](#) 15  
[CT\\_Ext complex type](#) 11  
[CT\\_IFilter complex type](#) 10  
[CT\\_indexedproperties complex type](#) 15  
[CT\\_merge complex type](#) 13  
[CT\\_Mime complex type](#) 11  
[CT\\_MimeMapping complex type](#) 11  
[CT\\_processor complex type](#) 19  
CT\_property complex type ([section 2.2.5.3](#) 13, [section 2.3.5.3](#) 16)  
  [CT\\_propset complex type](#) 16  
  [CT\\_select complex type](#) 13  
  [CT\\_Trust complex type](#) 10  
CustomPropertyExtractors.xml  
  [CT\\_extractor complex type](#) 30

[extensions](#) 30  
  [global attributes](#) 30  
  [global elements](#) 30  
  [overview](#) 30  
  [part enumerations](#) 30  
  [ST\\_dictionary\\_name simple type](#) 32  
  [ST\\_name simple type](#) 31  
  [ST\\_property simple type](#) 32  
  [ST\\_type simple type](#) 31  
  [ST\\_yield\\_values simple type](#) 32  
[CustomPropertyExtractors.xml example](#) 35  
[CustomPropertyExtractors.xml schema](#) 44

### D

Data types and fields - common ([section 2 9](#), [section 2 9](#))  
Details  
  common data types and fields ([section 2 9](#), [section 2 9](#))  
    [CT\\_dictionary complex type](#) 31  
    [CT\\_extractor complex type](#) 30  
    [CT\\_Mapping complex type](#) 22  
    [CT\\_MappingGroup complex type](#) 22  
    [CT\\_Namespace complex type](#) 21  
    [CT\\_SubTree complex type](#) 21  
    [CT\\_XMLMappings complex type](#) 21  
    [CT\\_category complex type](#) 15  
    [CT\\_Ext complex type](#) 11  
    [CT\\_IFilter complex type](#) 10  
    [CT\\_indexedproperties complex type](#) 15  
    [CT\\_merge complex type](#) 13  
    [CT\\_Mime complex type](#) 11  
    [CT\\_MimeMapping complex type](#) 11  
    [CT\\_processor complex type](#) 19  
    CT\_property complex type ([section 2.2.5.3](#) 13, [section 2.3.5.3](#) 16)  
    [CT\\_propset complex type](#) 16  
    [CT\\_select complex type](#) 13  
    [CT\\_Trust complex type](#) 10  
    [CustomPropertyExtractors.xml](#) 30  
    [ManagedProperties.xml](#) 12  
    [OptionalProcessing.xml](#) 18  
    [ST\\_paragraph-sep simple type](#) 23  
    [ST\\_path simple type](#) 24  
    [ST\\_propset simple type](#) 23  
    [ST\\_uri simple type](#) 24  
    [ST\\_active simple type](#) 19  
    [ST\\_attr simple type](#) 24  
    [ST\\_base-path simple type](#) 27  
    [ST\\_cpindexed simple type](#) 17  
    ST\_cpname simple type ([section 2.2.6.2](#) 14, [section 2.3.6.3](#) 17)  
    [ST\\_dictionary\\_name simple type](#) 32  
    [ST\\_discover simple type](#) 17  
    [ST\\_ignore-whitespace simple type](#) 25  
    [ST\\_indexed simple type](#) 16  
    [ST\\_mq-base-path simple type](#) 27  
    [ST\\_mq-mode simple type](#) 29



- [ST\\_mg-rec-post-str simple type](#) 28
- [ST\\_mg-rec-sep-str simple type](#) 28
- [ST\\_mg-select simple type](#) 28
- [ST\\_Mime simple type](#) 11
- [ST\\_MimeType simple type](#) 11
- [ST\\_mode simple type](#) 27
- [ST\\_m-propset simple type](#) 24
- [ST\\_m-type simple type](#) 24
- ST\_Name simple type ([section 2.1.6.3](#) 12, [section 2.2.6.1](#) 14, [section 2.3.6.1](#) 16, [section 2.4.6.1](#) 19, [section 2.5.6.3](#) 23, [section 2.5.6.19](#) 28, [section 2.6.6.1](#) 31)
- [ST\\_post-str simple type](#) 25
- [ST\\_pre-str simple type](#) 25
- [ST\\_property simple type](#) 32
- [ST\\_propset simple type](#) 18
- [ST\\_sep-str simple type](#) 25
- [ST\\_shallow simple type](#) 26
- [ST\\_strip-tags simple type](#) 26
- ST\_type simple type ([section 2.2.6.3](#) 14, [section 2.5.6.23](#) 29, [section 2.6.6.2](#) 31)
- [ST\\_yield\\_values simple type](#) 32
- [user\\_converter\\_rules.xml](#) 9
- [XMLMapper.xml](#) 20
- Details - CustomPropertyExtractors.xml
  - [extensions](#) 30
  - [global attributes](#) 30
  - [global elements](#) 30
  - [part enumerations](#) 30
- Details - ManagedProperties.xml
  - [extensions](#) 12
  - [global attributes](#) 13
  - [global elements](#) 12
  - [part enumerations](#) 12
- Details - OptionalProcessing.xml
  - [extensions](#) 18
  - [global attributes](#) 19
  - [global elements](#) 18
  - [part enumerations](#) 18
- [Details - PropertyCategories.xml](#) 14
  - [extensions](#) 15
  - [global attributes](#) 15
  - [global elements](#) 15
  - [part enumerations](#) 15
- Details - user\_converter\_rules.xml
  - [extensions](#) 10
  - [global attributes](#) 10
  - [global elements](#) 10
  - [part enumerations](#) 9
- Details - XMLMapper.xml
  - [extensions](#) 20
  - [global attributes](#) 21
  - [global elements](#) 20
  - [part enumerations](#) 20

## E

- Elements - global
  - [CustomPropertyExtractors.xml](#) 30
  - [ManagedProperties.xml](#) 12
  - [OptionalProcessing.xml](#) 18
  - [PropertyCategories.xml](#) 15

- [user\\_converter\\_rules.xml](#) 10
- [XMLMapper.xml](#) 20

## Examples

- [CustomPropertyExtractors.xml](#) 35
- [ManagedProperties.xml](#) 33
- [OptionalProcessing.xml](#) 34
- [PropertyCategories.xml](#) 33
- [user\\_converter\\_rules.xml](#) 34
- [XMLMapper.xml](#) 34

## Extensions

- [CustomPropertyExtractors.xml](#) 30
- [ManagedProperties.xml](#) 12
- [OptionalProcessing.xml](#) 18
- [PropertyCategories.xml](#) 15
- [user\\_converter\\_rules.xml](#) 10
- [XMLMapper.xml](#) 20

## F

- [Fields - vendor-extensible](#) 8

## Full XML schema

- [CustomPropertyExtractors.xsd](#) 44
- [ManagedProperties.xsd](#) 38
- [OptionalProcessing.xsd](#) 40
- [PropertyCategories.xsd](#) 39
- [user\\_converter\\_rules.xsd](#) 37
- [XMLMapper.xsd](#) 40

## G

### Global attributes

- [CustomPropertyExtractors.xml](#) 30
- [ManagedProperties.xml](#) 13
- [OptionalProcessing.xml](#) 19
- [PropertyCategories.xml](#) 15
- [user\\_converter\\_rules.xml](#) 10
- [XMLMapper.xml](#) 21

### Global elements

- [CustomPropertyExtractors.xml](#) 30
- [ManagedProperties.xml](#) 12
- [OptionalProcessing.xml](#) 18
- [PropertyCategories.xml](#) 15
- [user\\_converter\\_rules.xml](#) 10
- [XMLMapper.xml](#) 20

- [Glossary](#) 6

## I

- [Implementer - security considerations](#) 36

- [Informative references](#) 7

- [Introduction](#) 6

## L

- [Localization](#) 8

## M

- ManagedProperties.xml

- [CT\\_merge\\_complex\\_type](#) 13

- [CT\\_property\\_complex\\_type](#) 13

- [CT\\_select\\_complex\\_type](#) 13

- [extensions](#) 12
- [global attributes](#) 13
- [global elements](#) 12
- [overview](#) 12
- [part enumerations](#) 12
- [ST\\_cname simple type](#) 14
- [ST\\_name simple type](#) 14
- [ST\\_type simple type](#) 14
- [ManagedProperties.xml example](#) 33
- [ManagedProperties.xsd schema](#) 38

## N

- [Normative references](#) 6

## O

- OptionalProcessing.xml
  - [CT\\_dictionary complex type](#) 31
  - [CT\\_processor complex type](#) 19
  - [extensions](#) 18
  - [global attributes](#) 19
  - [global elements](#) 18
  - [overview](#) 18
  - [part enumerations](#) 18
  - [ST\\_active simple type](#) 19
  - [ST\\_name simple type](#) 19
- [OptionalProcessing.xml example](#) 34
- [OptionalProcessing.xsd schema](#) 40
- [Overview \(synopsis\)](#) 7

## P

- Part enumerations
  - [CustomPropertyExtractors.xml](#) 30
  - [ManagedProperties.xml](#) 12
  - [OptionalProcessing.xml](#) 18
  - [PropertyCategories.xml](#) 15
  - [user\\_converter\\_rules.xml](#) 9
  - [XMLMapper.xml](#) 20
- [Product behavior](#) 46
- PropertyCategories.xml
  - [CT\\_category complex type](#) 15
  - [CT\\_indexedproperties complex type](#) 15
  - [CT\\_property complex type](#) 16
  - [CT\\_propset complex type](#) 16
  - [extensions](#) 15
  - [global attributes](#) 15
  - [global elements](#) 15
  - [overview](#) 14
  - [part enumerations](#) 15
  - [ST\\_cpindexed simple type](#) 17
  - [ST\\_cname simple type](#) 17
  - [ST\\_discover simple type](#) 17
  - [ST\\_indexed simple type](#) 16
  - [ST\\_name simple type](#) 16
  - [ST\\_propset simple type](#) 18
- [PropertyCategories.xml example](#) 33
- [PropertyCategories.xsd schema](#) 39

## R

- [References](#) 6
  - [informative](#) 7
  - [normative](#) 6
- [Relationship to protocols and other structures](#) 7

## S

- Schema
  - [CustomPropertyExtractors.xsd](#) 44
  - [ManagedProperties.xsd](#) 38
  - [OptionalProcessing.xsd](#) 40
  - [PropertyCategories.xsd](#) 39
  - [user\\_converter\\_rules.xsd](#) 37
  - [XMLMapper.xsd](#) 40
  - [Security - implementer considerations](#) 36
- Simple types
  - [ST\\_paragraph-sep](#) 23
  - [ST\\_path](#) 24
  - [ST\\_propset](#) 23
  - [ST\\_uri](#) 24
  - [ST\\_active](#) 19
  - [ST\\_attr](#) 24
  - [ST\\_base-path](#) 27
  - [ST\\_cpindexed](#) 17
  - [ST\\_cname](#) ([section 2.2.6.2](#) 14, [section 2.3.6.3](#) 17)
  - [ST\\_dictionary\\_name](#) 32
  - [ST\\_discover](#) 17
  - [ST\\_ignore-whitespace](#) 25
  - [ST\\_indexed](#) 16
  - [ST\\_mq-base-path](#) 27
  - [ST\\_mq-mode](#) 29
  - [ST\\_mq-rec-post-str](#) 28
  - [ST\\_mq-rec-sep-str](#) 28
  - [ST\\_mq-select](#) 28
  - [ST\\_Mime](#) 11
  - [ST\\_MimeType](#) 11
  - [ST\\_mode](#) 27
  - [ST\\_m-propset](#) 24
  - [ST\\_m-type](#) 24
  - [ST\\_Name](#) ([section 2.1.6.3](#) 12, [section 2.2.6.1](#) 14, [section 2.3.6.1](#) 16, [section 2.4.6.1](#) 19, [section 2.5.6.3](#) 23, [section 2.5.6.19](#) 28, [section 2.6.6.1](#) 31)
  - [ST\\_post-str](#) 25
  - [ST\\_pre-str](#) 25
  - [ST\\_property](#) 32
  - [ST\\_propset](#) 18
  - [ST\\_sep-str](#) 25
  - [ST\\_shallow](#) 26
  - [ST\\_strip-tags](#) 26
  - [ST\\_type](#) ([section 2.2.6.3](#) 14, [section 2.5.6.23](#) 29, [section 2.6.6.2](#) 31)
  - [ST\\_yield\\_values](#) 32
  - [ST\\_paragraph-sep simple type](#) 23
  - [ST\\_path simple type](#) 24
  - [ST\\_propset simple type](#) 23
  - [ST\\_uri simple type](#) 24
  - [ST\\_active simple type](#) 19
  - [ST\\_attr simple type](#) 24
  - [ST\\_base-path simple type](#) 27
  - [ST\\_cpindexed simple type](#) 17

[ST\\_cpname simple type \(section 2.2.6.2 14, section 2.3.6.3 17\)](#)  
[ST\\_dictionary\\_name simple type 32](#)  
[ST\\_discover simple type 17](#)  
[ST\\_ignore-whitespace simple type 25](#)  
[ST\\_indexed simple type 16](#)  
[ST\\_mq-base-path simple type 27](#)  
[ST\\_mq-mode simple type 29](#)  
[ST\\_mq-rec-post-str simple type 28](#)  
[ST\\_mq-rec-sep-str simple type 28](#)  
[ST\\_mq-select simple type 28](#)  
[ST\\_Mime simple type 11](#)  
[ST\\_MimeType simple type 11](#)  
[ST\\_mode simple type 27](#)  
[ST\\_m-propset simple type 24](#)  
[ST\\_m-type simple type 24](#)  
[ST\\_Name simple type \(section 2.1.6.3 12, section 2.2.6.1 14, section 2.3.6.1 16, section 2.4.6.1 19, section 2.5.6.3 23, section 2.5.6.19 28, section 2.6.6.1 31\)](#)  
[ST\\_post-str simple type 25](#)  
[ST\\_pre-str simple type 25](#)  
[ST\\_property simple type 32](#)  
[ST\\_propset simple type 18](#)  
[ST\\_sep-str simple type 25](#)  
[ST\\_shallow simple type 26](#)  
[ST\\_strip-tags simple type 26](#)  
[ST\\_type simple type \(section 2.2.6.3 14, section 2.5.6.23 29, section 2.6.6.2 31\)](#)  
[ST\\_yield\\_values simple type 32](#)  
**Structures**  
[CustomPropertyExtractors.xml 30](#)  
[ManagedProperties.xml 12](#)  
[OptionalProcessing.xml 18](#)  
[overview \(section 2 9, section 2 9\)](#)  
[PropertyCategories.xml 14](#)  
[user\\_converter\\_rules.xml 9](#)  
[XMLMapper.xml 20](#)

## T

[Tracking changes 47](#)

**Types – CustomPropertyExtractors.xml**  
[CT\\_extractor complex type 30](#)  
[ST\\_dictionary\\_name simple type 32](#)  
[ST\\_name simple type 31](#)  
[ST\\_property simple type 32](#)  
[ST\\_type simple type 31](#)  
[ST\\_yield\\_values simple type 32](#)

**Types – ManagedProperties.xml**  
[CT\\_merge complex type 13](#)  
[CT\\_property complex type 13](#)  
[CT\\_select complex type 13](#)  
[ST\\_cpname simple type 14](#)  
[ST\\_name simple type 14](#)  
[ST\\_type simple type 14](#)

**Types – OptionalProcessing.xml**  
[CT\\_dictionary complex type 31](#)  
[CT\\_processor complex type 19](#)  
[ST\\_active simple type 19](#)  
[ST\\_name simple type 19](#)

**Types – PropertyCategories.xml**

[CT\\_category complex type 15](#)  
[CT\\_indexedproperties complex type 15](#)  
[CT\\_property complex type 16](#)  
[CT\\_propset complex type 16](#)  
[ST\\_cpindexed simple type 17](#)  
[ST\\_cpname simple type 17](#)  
[ST\\_discover simple type 17](#)  
[ST\\_indexed simple type 16](#)  
[ST\\_name simple type 16](#)  
[ST\\_propset simple type 18](#)

**Types – user\_converter\_rules.xml**

[CT\\_Ext complex type 11](#)  
[CT\\_IFilter complex type 10](#)  
[CT\\_Mime complex type 11](#)  
[CT\\_MimeMapping complex type 11](#)  
[CT\\_Trust complex type 10](#)  
[ST\\_Mime simple type 11](#)  
[ST\\_MimeType simple type 11](#)  
[ST\\_Name simple type 12](#)

**Types – XMLMapper.xml**

[CT\\_Mapping complex type 22](#)  
[CT\\_MappingGroup complex type 22](#)  
[CT\\_Namespace complex type 21](#)  
[CT\\_SubTree complex type 21](#)  
[CT\\_XMLMappings complex type 21](#)  
[ST\\_paragraph-sep simple type 23](#)  
[ST\\_path simple type 24](#)  
[ST\\_propset simple type 23](#)  
[ST\\_uri simple type 24](#)  
[ST\\_attr simple type 24](#)  
[ST\\_base-path simple type 27](#)  
[ST\\_ignore-whitespace simple type 25](#)  
[ST\\_mq-base-path simple type 27](#)  
[ST\\_mq-mode simple type 29](#)  
[ST\\_mq-rec-post-str simple type 28](#)  
[ST\\_mq-rec-sep-str simple type 28](#)  
[ST\\_mq-select simple type 28](#)  
[ST\\_mode simple type 27](#)  
[ST\\_m-propset simple type 24](#)  
[ST\\_m-type simple type 24](#)  
[ST\\_name simple type \(section 2.5.6.3 23, section 2.5.6.19 28\)](#)  
[ST\\_post-str simple type 25](#)  
[ST\\_pre-str simple type 25](#)  
[ST\\_sep-str simple type 25](#)  
[ST\\_shallow simple type 26](#)  
[ST\\_strip-tags simple type 26](#)  
[ST\\_type simple type 29](#)

## U

**user\_converter\_rules.xml**  
[CT\\_Ext complex type 11](#)  
[CT\\_IFilter complex type 10](#)  
[CT\\_Mime complex type 11](#)  
[CT\\_MimeMapping complex type 11](#)  
[CT\\_Trust complex type 10](#)  
[extensions 10](#)  
[global attributes 10](#)  
[global elements 10](#)  
[overview 9](#)  
[part enumerations 9](#)

[ST Mime simple type](#) 11  
[ST MimeType simple type](#) 11  
[ST Name simple type](#) 12  
[user\\_converter\\_rules.xml example](#) 34  
[user\\_converter\\_rules.xsd schema](#) 37

## V

[Vendor-extensible fields](#) 8  
[Versioning](#) 8

## X

### XML schema

[CustomPropertyExtractors.xsd](#) 44  
[ManagedProperties.xsd](#) 38  
[OptionalProcessing.xsd](#) 40  
[PropertyCategories.xsd](#) 39  
[user\\_converter\\_rules.xsd](#) 37  
[XMLMapper.xsd](#) 40

### XMLMapper.xml

[CT Mapping complex type](#) 22  
[CT MappingGroup complex type](#) 22  
[CT Namespace complex type](#) 21  
[CT SubTree complex type](#) 21  
[CT XMLMappings complex type](#) 21  
[extensions](#) 20  
[global attributes](#) 21  
[global elements](#) 20  
[overview](#) 20  
[part enumerations](#) 20  
[ST paragraph-sep simple type](#) 23  
[ST path simple type](#) 24  
[ST propset simple type](#) 23  
[ST uri simple type](#) 24  
[ST attr simple type](#) 24  
[ST base-path simple type](#) 27  
[ST ignore-whitespace simple type](#) 25  
[ST mg-base-path simple type](#) 27  
[ST mg-mode simple type](#) 29  
[ST mg-rec-post-str simple type](#) 28  
[ST mg-rec-sep-str simple type](#) 28  
[ST mg-select simple type](#) 28  
[ST mode simple type](#) 27  
[ST m-propset simple type](#) 24  
[ST m-type simple type](#) 24  
[ST\\_name simple type](#) ([section 2.5.6.3](#) 23, [section 2.5.6.19](#) 28)  
[ST post-str simple type](#) 25  
[ST pre-str simple type](#) 25  
[ST sep-str simple type](#) 25  
[ST shallow simple type](#) 26  
[ST strip-tags simple type](#) 26  
[ST type simple type](#) 29  
[XMLMapper.xml example](#) 34  
[XMLMapper.xsd schema](#) 40