

# ESA XML Use

---

XML messages are defined using the [W3C XML Schema](#) language. You can find this scheme within the sync.xsd example file in syncHome. All messages from or to Agiloft must be well-formed XML documents, starting with an XML declaration and a root <sync> tag. A complete, formal XML This ensures that Agiloft and the ESA will understand each other.

## Schematic Examples

---

This section contains examples of XML messages in the ESA:

### Calls to ESA

All messages are framed by <sync> and <esa-call> elements and are prepended by an XML declaration. Actual messages look like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync xsi:noNamespaceSchemaLocation="sync.xsd"
:xsi="http://www.w3.org/2001/XMLSchema-instance">
<esa-call call-id="12345">
<startSync>
<external-system-id>My-External-ID</external-system-id>
</startSync>
</esa-call>
</sync>
```

### Results from ESA

All result messages should also include the XML declaration and be wrapped into <result> tags, like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync xsi:noNamespaceSchemaLocation="sync.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<result response-to="12345">
<value>1.0</value>
...</result>
</sync>
```

# Calls to HelperAPI

All these messages must be framed by <sync> and <api-call> elements and are prepended by the XML declaration. Actual messages look like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync xsi:noNamespaceSchemaLocation="sync.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <api-call call-id="12345">
    <startSync asynchronous="true">
      <external-system-id>My-External-ID</external-system-id>
    </startSync>
  </api-call>
</sync>
```

## Results from API

These messages are identical in form to the ESA call results.

# ESA XML Message Schema

---

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!--Root element
is sync, having exactly one "call" or "result"-->

  <xs:element
name="sync">
  <xs:complexType>
  <xs:choice maxOccurs="1">
  <xs:sequence>
  <xs:element name="esa-call" type="esaCallType"/>
  </xs:sequence>
  <xs:element name="api-call" type="apiCallType"/>
  </xs:sequence>
  <xs:sequence>
  <xs:element name="result" type="resultType"/>
  </xs:sequence>
  </xs:choice>
  </xs:complexType>
  </xs:element>
```

```
<!--External System Adapter Calls-->
<xs:group name="esa-calls">
<xs:choice>
<xs:element name="startSync" type="startSyncEsa"/>
<xs:element name="leaseSession" type="noArgCall"/>
<xs:element name="endSync" type="noArgCall"/>
<xs:element name="release" type="noArgCall"/>
<xs:element name="getAllowedModes" type="noArgCall"/>
<xs:element name="needSyncAgain" type="noArgCall"/>
<xs:element name="getCurrentTime" type="noArgCall"/>
<xs:element name="configure" type="configure"/>
<xs:element name="getStructureList" type="localeOnly"/>
<xs:element name="getFieldList" type="getRecordMeta"/>
<xs:element name="getRelations" type="getRecordMeta"/>
<xs:element name="getCollections" type="getRecordMeta"/>
<xs:element name="getParametersMeta" type="localeOnly"/>
<xs:element name="getModified" type="getChanges"/>
<xs:element name="getModifiedPaged" type="getChanges"/>
<xs:element name="getDeleted" type="getChanges"/>
<xs:element name="getDeletedPaged" type="getChanges"/>
<xs:element name="leaseCursor" type="updateCursor"/>
<xs:element name="closeCursor" type="updateCursor"/>
<xs:element name="readDataPage" type="readDataPage"/>
<xs:element name="read" type="read"/>
<xs:element name="create" type="create"/>
<xs:element name="update" type="update"/>
<xs:element name="delete" type="delete"/>
<xs:element name="countRange" type="countRange"/>
<xs:element name="checkDelayedCreate" type="checkDelayedCreate"/>
<xs:element name="checkDelayedUpdate" type="checkDelayedUpdate"/>
<xs:element name="checkDelayedDelete" type="checkDelayedDelete"/>
<xs:element name="getProgressReport" type="noArgCall"/>
<xs:element name="getDetailedReport" type="noArgCall"/>
<xs:element name="syncErrorNotify" type="syncErrorNotify"/>
</xs:choice>
</xs:group>
```

```

<!--Helper API calls-->
<xs:group name="helper-api-calls">
<xs:choice>
<xs:element name="startSync" type="startSyncApi"/>
<xs:element name="getParameter" type="getParameter"/>
<xs:element name="getPollPeriod" type="getPollPeriod"/>
<xs:element name="trackRecordDeletion" type="trackRecordDeletion"/>
<xs:element name="detectDeleted" type="detectDeleted"/>
<xs:element name="isKnownID" type="isKnownID"/>
<xs:element name="enumerateKnownIDs" type="enumerateKnownIDs"/>
</xs:choice>
</xs:group>

```

```

<!--Calls-->
<xs:complexType name="callType" abstract="true">
<xs:attribute name="call-id" use="required" type="xs:nonNegativeInteger"/>
</xs:complexType>

<xs:complexType name="esaCallType">
<xs:complexContent>
<xs:extension base="callType">
<xs:group ref="esa-calls"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="apiCallType">
<xs:complexContent>
<xs:extension base="callType">
<xs:group ref="helper-api-calls"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!--Esa.leaseSession, Esa.getStructureList, Esa.endSync, Esa.release, Esa.
getAllowedRunModes, Esa.blockMappedFieldsChangesEsa.needSyncAgain, Esa.
getCurrentTime has empty parameters -->
<xs:complexType name="noArgCall">
</xs:complexType>

<!--External System Adapter calls-->

<!--Start Sync-->
<xs:complexType name="startSyncEsa">
<xs:sequence>
<xs:element name="external-system-id" type="xs:string"/>
</xs:sequence>
</xs:complexType>

```

```
<!--Esa.getFieldList, Esa.getRelations, Esa.getCollections -->
<xs:complexType name="getRecordMeta">
  <xs:sequence>
    <xs:element name="structure" type="xs:string"/>
    <xs:element name="locale" type="xs:language"/>
  </xs:sequence>
</xs:complexType>

<!--Esa.getParametersMeta, Esa.getStructures -->
<xs:complexType name="localeOnly">
  <xs:sequence>
    <xs:element name="locale" type="xs:language"/>
  </xs:sequence>
</xs:complexType>

<!--Esa.configure-->
<xs:complexType name="configure">
  <xs:sequence>
    <xs:element name="external-system-id" type="xs:string"/>
    <xs:element name="force" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>

<!--Esa.getModified, Esa.getModifiedPaged, Esa.getDeleted, Esa.getDeletedPaged -->
<xs:complexType name="getChanges">
  <xs:sequence>
    <xs:element name="structure" type="xs:string"/>
    <xs:element name="after" type="xs:dateTime" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!--Esa.leaseCursor, Esa.closeCursor -->
<xs:complexType name="updateCursor">
  <xs:sequence>
    <xs:element name="cursor-id" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<!--Esa.readDataPage -->
<xs:complexType name="readDataPage">
  <xs:sequence>
    <xs:element name="cursor-id" type="xs:string"/>
    <xs:element name="page-index" type="xs:nonNegativeInteger" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!--Esa.read-->
<xs:complexType name="read">
  <xs:sequence>
    <xs:element name="structure" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:element name="id" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<!--Esa.create-->
<xs:complexType name="create">
<xs:sequence>
<xs:element name="structure" type="xs:string"/>
<xs:element name="record" type="recordType"/>
</xs:sequence>
</xs:complexType>

<!--Esa.update-->
<xs:complexType name="update">
<xs:sequence>
<xs:element name="structure" type="xs:string"/>
<xs:element name="last-seen" type="xs:dateTime" id="last-seen-update"/>
<xs:element name="record" type="recordType"/>
</xs:sequence>
</xs:complexType>

<!--Esa.delete-->
<xs:complexType name="delete">
<xs:sequence>
<xs:element name="structure" type="xs:string"/>
<xs:element name="last-seen" id="last-seen-delete" type="xs:dateTime"/>
<xs:element name="id" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<!--Esa.countRange -->
<xs:complexType name="countRange">
<xs:sequence>
<xs:element name="structure" type="xs:string"/>
<xs:element name="min-id" type="xs:string"/>
<xs:element name="max-id" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<!--Esa.checkDelayedCreate -->
<xs:complexType name="checkDelayedCreate">
<xs:sequence>
<xs:element name="structure" type="xs:string"/>
<xs:element name="token" type="xs:string"/>
</xs:sequence>
</xs:complexType>

<!--Esa.checkDelayedUpdate -->
<xs:complexType name="checkDelayedUpdate">
<xs:sequence>
<xs:element name="structure" type="xs:string"/>

```

```
<xs:element name="token" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<!--Esa.syncErrorNotify -->
<xs:complexType name="syncErrorNotify">
<xs:sequence>
<xs:element name="error" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<!--Helper API-->
```

```
<!--Start Sync-->
<xs:complexType name="startSyncApi">
<xs:sequence>
<xs:element name="external-system-id" type="xs:string"/>
</xs:sequence>
<xs:attribute name="asynchronous" use="required" type="xs:boolean"/>
</xs:complexType>
```

```
<!--HelperApi.getParameter -->
<xs:complexType name="getParameter">
<xs:sequence>
<xs:element name="external-system-id" type="xs:string"/>
<xs:element name="name" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<!--HelperApi.getPollPeriod -->
<xs:complexType name="getPollPeriod">
<xs:sequence>
<xs:element name="external-system-id" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<!--HelperApi.trackRecordDeletion -->
<xs:complexType name="trackRecordDeletion">
<xs:sequence>
<xs:element name="external-system-id" type="xs:string"/>
<xs:element name="structure" type="xs:string"/>
<xs:element name="id" type="xs:string"/>
<xs:element name="time" type="xs:dateTime" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
```

```
<!--HelperApi.detectDeleted -->
<xs:complexType name="detectDeleted">
<xs:sequence>
```



```
<xs:element name="external-system-id" type="xs:string"/>
<xs:element name="after" type="xs:dateTime"/>
<xs:element name="structure" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<!--HelperApi.isKnownID -->
<xs:complexType name="isKnownID">
<xs:sequence>
<xs:element name="external-system-id" type="xs:string"/>
<xs:element name="structure" type="xs:string"/>
<xs:element name="id" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<!--HelperApi.enumerateKnownIDs -->
<xs:complexType name="enumerateKnownIDs">
<xs:sequence>
<xs:element name="external-system-id" type="xs:string"/>
<xs:element name="known-before" type="xs:dateTime"/>
<xs:element name="structure" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<!--Data types-->
<!--External Record-->
<xs:complexType name="recordType">
<xs:sequence>

<!--Field values-->
<xs:element name="field" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

<!--Related record IDs-->
<xs:element name="relation" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="related" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
```

```

<!--Collections-->
<xs:element name="collection" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="record" type="recordType" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>

<!--Record Attributes-->
<xs:attribute name="id" type="xs:string" use="required"/>
<xs:attribute name="old-id" type="xs:string"/>
<xs:attribute name="modified" type="xs:dateTime"/>
<xs:attribute name="new" type="xs:boolean"/>
</xs:complexType>

<!--External Structure Descriptor-->
<xs:complexType name="structureType">
<xs:attribute name="name" type="xs:string"/>
<xs:attribute name="screen-name" type="xs:string"/>
</xs:complexType>

<!--Exception Descriptor-->
<xs:simpleType name="exceptionTypeEnum">
<xs:restriction base="xs:string">
<xs:enumeration value="general"/>
<xs:enumeration value="record"/>
<xs:enumeration value="recorddelayed"/>
<xs:enumeration value="configuration"/>
<xs:enumeration value="alreadyconfigured"/>
<xs:enumeration value="optlockfailed"/>
<xs:enumeration value="concurrentdelete"/>
<xs:enumeration value="skipstructure"/>
</xs:restriction>
</xs:simpleType>
<xs:complexType name="exceptionType">
<xs:sequence>
<xs:element name="message" type="xs:string"/>
<xs:element name="trace" type="xs:string"/>
<xs:element name="configured-to" type="xs:string" minOccurs="0"/>
<xs:element name="external-id" type="xs:string" minOccurs="0"/>
<xs:element name="modified-at" type="xs:dateTime" minOccurs="0"/>
<xs:element name="delay-token" type="xs:string" minOccurs="0"/>
<xs:element name="structure-name" type="xs:string" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="type" type="exceptionTypeEnum" use="required"/>
</xs:complexType>

<!--Enumerated (field or parameter) value-->

```

```

<xs:complexType name="enumValueType">
  <xs:sequence>
    <xs:element name="value" type="xs:string"/>
    <xs:element name="screen-name" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<!--External Field Descriptor-->
<xs:simpleType name="externalFieldTypeEnum">
  <xs:restriction base="xs:string">
    <!--Supported Schema types-->
    <xs:enumeration value="ew:enum"/>
    <xs:enumeration value="ew:attachedFiles"/>
    <xs:enumeration value="xsd:string"/>
    <xs:enumeration value="xsd:normalizedString"/>
    <xs:enumeration value="xsd:token"/>
    <xs:enumeration value="xsd:language"/>
    <xs:enumeration value="xsd:duration"/>
    <xs:enumeration value="xsd:date"/>
    <xs:enumeration value="xsd:dateTime"/>
    <xs:enumeration value="xsd:time"/>
    <xs:enumeration value="xsd:gYearMonth"/>
    <xs:enumeration value="xsd:gYear"/>
    <xs:enumeration value="xsd:gMonthDay"/>
    <xs:enumeration value="xsd:gDay"/>
    <xs:enumeration value="xsd:gMonth"/>
    <xs:enumeration value="xsd:boolean"/>
    <xs:enumeration value="xsd:base64Binary"/>
    <xs:enumeration value="xsd:hexBinary"/>
    <xs:enumeration value="xsd:float"/>
    <xs:enumeration value="xsd:decimal"/>
    <xs:enumeration value="xsd:double"/>
    <xs:enumeration value="xsd:byte"/>
    <xs:enumeration value="xsd:unsignedByte"/>
    <xs:enumeration value="xsd:int"/>
    <xs:enumeration value="xsd:unsignedInt"/>
    <xs:enumeration value="xsd:integer"/>
    <xs:enumeration value="xsd:negativeInteger"/>
    <xs:enumeration value="xsd:nonNegativeInteger"/>
    <xs:enumeration value="xsd:positiveInteger"/>
    <xs:enumeration value="xsd:nonPositiveInteger"/>
    <xs:enumeration value="xsd:long"/>
    <xs:enumeration value="xsd:unsignedLong"/>
    <xs:enumeration value="xsd:short"/>
    <xs:enumeration value="xsd:unsignedShort"/>
    <xs:enumeration value="xsd:anyURI"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="externalFieldType">
  <xs:sequence>
    <xs:element name="enum-value" minOccurs="0" maxOccurs="unbounded" type="

```

```

enumValueType"/>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="screen-name" type="xs:string" use="required"/>
<xs:attribute name="required" type="xs:boolean" use="required"/>
<xs:attribute name="identifying" type="xs:boolean" use="required"/>
<xs:attribute name="updatable" type="xs:boolean" use="required"/>
<xs:attribute name="updatableOnCreate" type="xs:boolean" use="required"/>
<xs:attribute name="max-length" type="xs:integer" use="optional" default="-1"/>
<xs:attribute name="type" type="externalFieldTypeEnum" use="required"/>
</xs:complexType>

<!--External Relation Descriptor-->
<xs:complexType name="externalRelationType">
<xs:attribute name="id" type="xs:string" use="required"/>
<xs:attribute name="related" type="xs:string" use="required"/>
<xs:attribute name="screen-name" type="xs:string" use="required"/>
<xs:attribute name="multiple" type="xs:boolean" use="required"/>
<xs:attribute name="required" type="xs:boolean" use="required"/>
<xs:attribute name="postponable" type="xs:boolean" use="required"/>
</xs:complexType>

<!--External Collection Descriptor-->
<xs:complexType name="externalCollectionType">
<xs:attribute name="id" type="xs:string" use="required"/>
<xs:attribute name="screen-name" type="xs:string" use="required"/>
<xs:attribute name="required" type="xs:boolean" use="required"/>
<xs:attribute name="updatable" type="xs:boolean" use="required"/>
<xs:attribute name="updatableOnCreate" type="xs:boolean" use="required"/>
</xs:complexType>

<!--Cursor Descriptor-->
<xs:complexType name="cursorType">
<xs:attribute name="id" type="xs:string" use="required"/>
<xs:attribute name="page-size" type="xs:positiveInteger" use="required"/>
<xs:attribute name="number-of-pages" type="xs:nonNegativeInteger" use="required"/>
</xs:complexType>

<!--ESA Parameter Meta-->
<xs:simpleType name="esaParameterTypeEnum">
<xs:restriction base="xs:string">
<xs:enumeration value="single"/>
<xs:enumeration value="radio"/>
<!--<xs:enumeration value="multi"/>-->
<!--<xs:enumeration value="expandable"/>-->
<xs:enumeration value="xml"/>
<xs:enumeration value="custom"/>
<xs:enumeration value="password"/>
<xs:enumeration value="timezone"/>
</xs:restriction>
</xs:simpleType>

```

```

<xs:simpleType name="esaParameterItemTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="text"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="float"/>
    <xs:enumeration value="boolean"/>
    <xs:enumeration value="date"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="parametersMetaType">
  <xs:sequence>
    <xs:element name="label" type="xs:string"/>
    <xs:element name="hint" type="xs:string"/>
    <xs:choice>
      <xs:element name="render-code" type="xs:string" minOccurs="0"/>
      <xs:sequence>
        <xs:element name="item-type" type="esaParameterItemTypeEnum"/>
        <xs:element name="default-value" minOccurs="0" type="xs:string"/>
        <xs:element name="enum-value" minOccurs="0" maxOccurs="unbounded" type="enumValueType"/>
      </xs:sequence>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="type" type="esaParameterTypeEnum" use="required"/>
  <xs:attribute name="required" type="xs:boolean" use="required"/>
</xs:complexType>
<xs:complexType name="parameterValueType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="type" type="esaParameterItemTypeEnum"/>
      <!--<xs:attribute name="valueScreenName" type="xs:string" use="optional"/>-->
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

```

<!--Result-->

```

```

<xs:complexType name="resultType">
  <xs:choice minOccurs="0"> <!-- Result can be empty as well -->

```

```

    <!--Simple values -->
    <xs:sequence>
      <xs:element name="value" type="xs:string"/>
    </xs:sequence>

```

```

    <!--Exception Result-->
    <xs:sequence>
      <xs:element name="exception" type="exceptionType"/>

```

```

</xs:sequence>

<!--Record list-->
<xs:sequence>
<xs:element name="record" type="recordType" maxOccurs="unbounded"/>
</xs:sequence>

<!--Structure list-->
<xs:sequence>
<xs:element name="structure" type="structureType" maxOccurs="unbounded"/>
</xs:sequence>

<!--Fields list-->
<xs:sequence>
<xs:element name="field" type="externalFieldType" maxOccurs="unbounded"/>
</xs:sequence>
  <!--Relations list-->
  <xs:sequence>
  <xs:element name="relation" type="externalRelationType" maxOccurs="unbounded"/>
  </xs:sequence>

<!--Collections list-->
<xs:sequence>
<xs:element name="collection" type="externalCollectionType" maxOccurs="unbounded"/>
</xs:sequence>

<!--Cursor result-->
<xs:sequence>
<xs:element name="cursor" type="cursorType"/>
</xs:sequence>

<!--ID list-->
<xs:sequence>
<xs:element name="id" maxOccurs="unbounded" type="xs:string">
</xs:element>
</xs:sequence>

<!--ESA Parameter Metas list -->
<xs:sequence>
<xs:element name="esa-parameter" type="parametersMetaType" maxOccurs="unbounded"/>
</xs:sequence>

<!--Value list -->
<xs:sequence>
<xs:element name="parameter-value" type="parameterValueType" maxOccurs="unbounded"/>
</xs:sequence>

<!--Simple values -->
<xs:sequence>
<xs:element name="nullValue" type="xs:boolean"/>
</xs:sequence>

```

```
</xs:choice>
<xs:attribute name="response-to" use="required" type="xs:nonNegativeInteger"/>
</xs:complexType>
</xs:schema>
```

## Data Mapped to Snippets

---

To assist in the coding of messages, many kinds of data are mapped to XML snippets associated with the ESA XML messages schema as follows:

## External Records

External records are mapped to XML snippets with the form:

```
<record id="id-value"
old-id="old-id-value"
modified="modificationtimestamp">
  <field name="field-name"> field value </field>
  ...
  <field name="field-name">...</field>
  <relation id="relation-id">
    <related>related-object-id</related>
    ...
    <related>related-object-id</related>
  </relation>
  ...
  <relation id="relation-id">
    ...
  </relation>
</collection>
<record> tags, describing collection members.
</collection>
</record>
```

## External Structures

External Structures are mapped to XML snippets with the form:

```
<structure name="name-value" screen-name="screen-name"/>
```

## External Collections

External Collections are mapped to XML snippets with the form:

```
<collection id="collection-id"
screen-name="screen-name"
required="boolean-value"
updatable="boolean-value"
updatableOnCreate="boolean-value"/>
```

## External Fields

External Fields are mapped to XML snippets with the form:

```
<field name="name-value"
screen-name="screen-name"
type="xsd-type-name"
required="true/false"
identifying="true/false"
updatable="true/false"
updatableOnCreate=" true/false"
maxLength="maximum length of string representation">
<!-- If a field is of enumeration type, it should have type "ew:enum"
and contain one or more nested <enum-value> elements: -->
<enum-value>
<value>value</value>
<screen-name>
name to be used in mapping editor
</screen-name>
</enum-value>
</field>
```



# External Relations

External Relations are mapped to XML snippets with the form:

```
<relation id="relation-id"
related="related-structure-name"
screen-name="screen-name"
multiple="Boolean-value"
required="Boolean-value"/>
```

# ESA Parameters Meta

The ESA parameters metadata are mapped to XML snippets with the form:

```
<esa-parameter
name="string-name"
type="single | radio | multi | expandable | xml"
required="true | false">
<label>Label to show above input</label>
<hint>Hint to show on the left</hint>
<item-type>
xsd:string | xsd:integer | xsd:double | xsd:Boolean
</item-type>
<default-value>
String-value | Integer-value | Double-value | Boolean-value
</default-value>
<!-- For parameters of type "radio", "multi" and
"expandable", a number of nested <enum-value>
elements should follow: -->
<enum-value>
<value>value</value>
<screen-name>
name to be used in mapping editor
</screen-name>
</enum-value>
</esa-parameter>
```

# Cursors

Cursors are mapped to XML snippets with the form:

```
<cursor id="cursor-id"  
page-size="records-per-page"  
number-of-pages="number-of-data-pages" />
```