

EDGAR[®] XFDL

Technical Specification



ELECTRONIC DATA GATHERING, ANALYSIS, AND RETRIEVAL

Version 2
November 2005

TABLE OF CONTENTS

1. PURPOSE	1-1
1.1 Updates	1-2
1.2 Changes by Template.....	1-2
2. ORGANIZATION OF THIS SPECIFICATION	2-1
3. XFDL TEMPLATES	3-1
4. FILER-CONSTRUCTED FILE FORMAT	4-1
5. FILER-CONSTRUCTED XFDL ELEMENTS	5-1
5.1 Version Nodes.....	5-1
5.2 End XFDL Node.....	5-1
5.3 Page Nodes.....	5-1
5.4 Valid Scope (Node) Identifiers	5-3
5.4.1 Valid EDGAR Scope Identifiers (<i>sids</i>).....	5-3
5.5 Submission Data Nodes	5-4
5.6 Field Nodes	5-5
5.7 Checkbox Nodes	5-5
5.8 Radio Button Nodes.....	5-6
5.9 Popup Nodes	5-6
5.10 Combobox Nodes.....	5-6
5.11 Data Nodes – (Enclosed Documents)	5-6
6. VERIFYING FILER-CONSTRUCTED STRUCTURE	6-1
7. HOW TO USE THE APPENDICES	7-1
8. SAMPLE FORMS	8-1
8.1 Sample 24F-2NT from Subtemplate1.xfd.....	8-1
8.2 Sample S-1 from Subtemplate1.xfd.....	8-2
8.3 Sample N-1 from Subtemplate1.xfd	8-3
8.4 Sample 144 submission from Subtemplate2.xfd	8-4
8.5 Sample PREM14A submission from Subtemplate2.xfd.....	8-5
8.6 Sample 10-12B from Subtemplate3.xfd	8-6
8.7 Sample CORRESP from Subtemplate4.xfd.....	8-7
8.8 Sample MODULE from Subtemplate5.xfd	8-8
8.9 Sample BULK from Subtemplate6.xfd	8-9

1. PURPOSE

The EDGARLink® portion of the Electronic Data Gathering, Analysis, and Retrieval (EDGAR) web-based filing environment is based on the PureEdge (formerly UWI.COM) Extensible Forms Description Language (XFDL) forms specification, which is an extension of the Extensible Markup Language (XML) specification. An EDGAR filer, from his or her client machine, must use the PureEdge Viewer application to open an XFDL Template and construct a file that he or she can then submit via the EDGAR Filing Website. The EDGAR server-side software uses the PureEdge Application Programming Interface (API) to parse the pertinent submission data from the filer's XFDL file.

XFDL forms are XML documents; the form definition is encoded using XML elements and attributes. The XFDL specification uses Document Type Definition (DTD) notation to express nesting and sequence relationships between the elements and attributes; while the constraints on element contents and attribute values are given in the Backus Naur Form (BNF) notation found in the XML specification.

Roughly 99 percent of the XFDL content of each EDGAR Template, as well as any submission file saved from the Template, is needed only for the rendering of the Graphical User Interface (GUI). The EDGAR submission processing software is concerned with the one percent of the XFDL content that contains the submission-related data. All the GUI-related items are ignored.

This document provides a specification for creating XFDL filings without the use of the PureEdge Viewer. The expectation is that software developers, working on behalf of filers, will construct software that will generate submissions that can be successfully parsed by EDGAR. "Filer-Constructed" is the term used to describe submissions compliant with this specification.¹

The benefits of filer-constructed submissions are twofold. First, these smaller files require less bandwidth for upload to the EDGAR Filing Website and therefore should upload quicker. Second, the server-side parser can process the filing quicker since all the GUI-related content is not in the file.

This specification, by itself, is not sufficient to generate correct filer-constructed submissions. We assume the reader is already familiar with XML, the various SEC form types, filing act requirements, and SEC syntax rules. The following documents provide additional filing information and provide a resource for the XFDL specification.

- The EDGAR Filer Manual may be obtained from <http://www.sec.gov>.
- The Extensible Forms Definition Language (XFDL) 4.0 Specification may be obtained from <http://www.w3.org/TR/1998/NOTE-XFDL-19980902.html>.

NOTE: Although the PureEdge product suite was used to develop the EDGAR electronic forms, PureEdge did not participate in this process. Do NOT contact PureEdge for information specific to the SEC form templates or this technical specification.

¹ The specific submission types that this specification supports are called out in Appendix I. Note that there may be other "Filer-Constructed" specifications for form types not supported here. For example, the EDGAR Ownership XML Technical Specification covers Section 16 Forms 3, 4, and 5. XFDL versions of those forms are not supported by EDGAR.

1.1 Updates

This revision of the XFDL Technical Specification includes updates to Submission Templates 1, 2, and 3 and the EDGARLink software. These updates are based on SEC requests and rulemaking activities as implemented in EDGAR Release 9.2. EDGAR 9.2 server software supports all of the field identifiers that were valid in the 9.1 version of the PureEdge templates.

1.2 Changes by Template

The following changes are called out in the appendices. Please refer to the appendices for details.

Template 1

1. The following additional submission types are allowed in the SubTable_submissionType_ field and the SubOffset_offsetForm_ field:
S-3ASR, F-3ASR, POSASR, 424B1, 424B2, 424B3, 424B4, 424B5, 424B7, 424B8
2. In order to support a new fee payment plan option for Well-Known Seasoned Issuers, the new SubOffering_typeOfPayment_ field is allowed for the following submission types:
S-3ASR, F-3ASR, POSASR, 424B1, 424B2, 424B3, 424B4, 424B5, 424B7, 424B8
3. In order to support fee-bearing takedowns, the new SubTable_feeTableInProspectusFlag_ field is allowed by Well-Known Seasoned Issuers for the following submission types:
424B1, 424B2, 424B3, 424B4, 424B5, 424B7, 424B8

Template 2

1. The following additional submission type is allowed in the SubTable_submissionType_ field:
FWP
2. The following additional submission types are allowed in the SubOffset_offsetForm_ field:
S-3ASR, F-3ASR, POSASR, 424B1, 424B2, 424B3, 424B4, 424B5, 424B7, 424B8
3. In order to specify that a filing is covered by Rule 163, the new SubTable_rule163Flag_ field is allowed for the following submission types:
FWP, 425
4. In order to specify that a filing is covered by Rule 433, the new SubTable_rule433Flag_ field is allowed for the following submission types:
FWP, 425
5. In order to specify that a filing is the first filing for an Issuing Entity, the new SubTable_firstFilingIssuingEntityFlag_ field and the SubTable_depositorCik_ and SubTable_depositorFileNumber_ fields are allowed for the following submission type:
FWP

Template 3

1. The following submission types were moved to Template 1 and are no longer allowed in the SubTable_submissionType_ field:
424B1, 424B2, 424B3, 424B4, 424B5, 424B7
2. In order to specify that a filer is a Voluntary Filer or a Well-Known Seasoned Issuer, the new SubFiler_voluntaryFilerFlag_ and SubFiler_wellKnownSeasonedIssuerFlag_ fields are required for the following submission types:
10-K, 10-K/A, 10-KT, 10-KT/A, 20-F, 20-F/A
3. In order to specify that a filer is a Shell Company, the new SubFiler_shellCompanyFlag_ field is required for the following submission types:
10-K, 10-K/A, 10-KT, 10-KT/A, 10KSB, 10KSB/A, 20-F, 20-F/A
4. The following additional item is allowed in the SubItem_itemId_ field:
5.06

2. ORGANIZATION OF THIS SPECIFICATION

The remaining sections of this specification provide the framework for successfully constructing a “filer-constructed” XFDL-based submission.

- Chapter 3, XFDL Templates, provides an overview of the six XFDL templates to which this specification applies.
- Chapter 4, Filer-Constructed File Format, provides general rules concerning the XFDL file being created.
- Chapter 5, Filer-Constructed XFDL Elements, details the specific rules for each type of XFDL node used to pass data to EDGAR.
- Chapter 6, Verifying Filer-Constructed Structure, describes a means of verifying that the structure of an XFDL file is correct. For example, you can use the described technique to find a missing end tag.
- Chapter 7, How to use the Appendices, describes the many appendices to this specification and how to use them. These appendices provide all the details concerning how to create a specific SEC form, which templates create which form types, and the set of valid values for several input fields.
- Chapter 8, Sample Forms, provides several examples (at least one for each template) of filer-constructed submissions.

3. XFDL TEMPLATES

The six XFDL templates to which this EDGAR XFDL Technical Specification applies are:

EDGAR Template Name	Filename on the EDGAR Filing Website	Purpose
Template1	Subtemplate1.xfd	117 submission types
Template2	Subtemplate2.xfd	121 submission types
Template3	Subtemplate3.xfd	166 submission types
Template4	Subtemplate4.xfd	CORRESPONDENCE submission
Template5	Subtemplate5.xfd	MODULE or SEGMENT submission
Template6	Subtemplate6.xfd	BULK submission

Template6 allows a filer to collect multiple XFDL files created with the other five templates into a single bundle for a BULK submission. The only constraint here is that one BULK submission cannot include another BULK submission.

The controlled master versions of these templates that are available for download on the EDGAR Filing Website. The master templates are saved in a PureEdge proprietary compressed file format to improve file upload and download efficiency and as a measure of security. As such, they cannot be viewed or manipulated without the PureEdge Viewer. All submission files created via the Viewer and the EDGAR templates are also saved in this same proprietary compressed file format.

In addition, each template is composed of multiple “pages” denoted within the template by XFDL page nodes. Section 5.3, Page Nodes, describes the standard page nodes that apply to each template.

4. FILER-CONSTRUCTED FILE FORMAT

Filer-constructed submissions must conform to the following file formatting requirements.

- a. A filer-constructed XFDL submission file must be ASCII text in either DOS or UNIX format.

That is, lines must be terminated with either a carriage return and line feed (`\r\n`) or with just a line feed (`\n`). The PureEdge API does not support the Macintosh format of just a carriage return (`\r`). Do not submit files with only carriage returns as end-of-line delineators.

- b. Because the PureEdge compression methodology is proprietary, you can only compress a filer-constructed submission by using the PureEdge Designer product.

When you use the Designer to compress a submission, use ASCII compression rather than binary.

- c. A filer-constructed submission file cannot be digitally signed.
- d. A filer-constructed submission file name must end with an `.xfd`, `.xfdl`, or `.frm` extension.

We recommend the `.xfd` extension, since several commercial software products use the `.frm` file type (for example, Visual Basic). Under Microsoft Windows, a user may find the `.frm` file type already registered with another software program on his workstation.

- e. The names of attached document files must follow the internal EDGAR file naming convention. The file naming convention is:

- File names can be no longer than 32 characters, including the file extension.
- The following are valid characters:
 - **Lowercase** letters
 - Digits 0-9
 - Up to one underscore
 - Up to one hyphen
 - Up to one period
- The first character must be a letter.
- Space characters are not allowed.
- Each file name must have a file type extension.
- File extensions must be `.txt`, `.htm`, `.pdf`, `.fil`, `.jpg`, `.gif`, `.xml`, and `.xsd`.

- f. Module and segment names referenced on PAGE4 of submissions must follow the internal EDGAR module/segment naming convention. The module/segment naming convention is:

- The module/segment name must be no longer than 15 characters.
- Valid characters are:
 - **Uppercase** letters
 - Digits 0-9
 - Underscores
 - Hyphens (not periods)

- The first character must be a letter.
- Space characters are not allowed.

g. Date values can be specified in any of the following formats:

- MM/DD/YY 10/10/01
- MM/DD/YYYY 10/10/2001
- MM-DD-YY 10-10-01
- MM-DD-YYYY 10-10-2001
- MMM DD, YY October 10, 01
- MMM DD, YYYY October 10, 2001

5. FILER-CONSTRUCTED XFDL ELEMENTS

An XFDL scope identifier, or *sid*, uniquely identifies an element within the scope of its logical parent. An XFDL element may have a *sid* attribute which uniquely identifies the form within a system of forms in a large deployment. A form may be composed of pages. Each page element must have a *sid* attribute that uniquely identifies the page within the surrounding XFDL form element. Pages may be composed of items. An item element must have a *sid* attribute that uniquely identifies the item within the surrounding page element. The minimal set of XFDL elements required to create filer-constructed submissions are discussed in the following sections. Other XFDL elements used in the master EDGAR XFDL templates, such as those for GUI rendering, are not discussed in this specification, and should not be provided when creating a filer-constructed submission.

NOTE: The following filer-constructed element definitions use example data to illustrate typical values. Remember to change these values as required when creating your filer-constructed submissions. If EDGAR restricts the value of a particular node, the list of valid values is provided.

5.1 Version Nodes

Two version tags must be included as the first two lines of each submission. These tags should not appear elsewhere in the submission. The current valid values are:

```
<?xml version="1.0"?>
<XFDL version="5.0.0">
```

5.2 End XFDL Node

To properly end a submission, this tag must be the last line of the submission. This tag should not appear elsewhere in the submission.

```
</XFDL>
```

5.3 Page Nodes

The `<page>` nodes are critical to EDGAR processing and must be included. The following table describes the data provided on each page and provides a mapping of pages to EDGARLink templates. An “x” in the cell indicates the page node is present in that template.

Page	Description	Template1	Template2	Template3	Template4 (Corre- spondence)	Template5 (Module/ Segment)	Template6 (Bulk)
1	Submission Header Data	x	x	x	x	x	
2	Enclosed Documents	x	x	x	x	x	x
3	Notification Data	x	x	x	x	x	
4	Module/ Segment References	x	x	x	x		
5	Offset Payments Data	x	x				
6	Offering Data	x	x				
7	Sales Shares Data	x					

The rules governing the <page> nodes are:

- a. Valid values for a <page> *sid* are “PAGE1”, “PAGE2”, “PAGE3”, “PAGE4”, “PAGE5”, “PAGE6”, and “PAGE7”.
- b. Pages must appear in sequential order, starting at PAGE1.
- c. All nodes except the two version nodes and the end XFDL node described in Sections 5.1 and 5.2 must be bounded by <page sid= > </page> nodes.
- d. You may completely exclude a page if none of the fields on that page contain any data.

For example, if an S-1 submission has no module or segment references, the filer-constructed submission does not need to contain a PAGE4.

5.4 Valid Scope (Node) Identifiers

This section lists the valid node, or scope (*sid*), identifiers. Only these *sids* may be included in a submission. If any other *sid* is detected during processing, EDGAR will suspend the submission.

5.4.1 Valid EDGAR Scope Identifiers (*sids*)

The following is a list of valid filer-constructed EDGAR scope identifiers. Appendix II provides a more detailed breakout of the valid *sids* by Template and Page. The table provided here is an alphabetical listing of the valid *sids*.

ModSegReferenceData_ccc_	SubModSeg_ModSegDescription_
ModSegReferenceData_cik_	SubModSeg_ModSegFileName_
ModSegReferenceData_documentType_	SubModSeg_ModSegMasterCIK_
ModSegReferenceData_modOrSegFlag_	SubModSeg_ModSegName_
ModSegReferenceData_nickName_	SubModSeg_ModSegType_
SubCompany_fileNumber_	SubOffering_aggregatePrice_
SubCompany_filerId_	SubOffering_offeringAmountPerShare_
SubCompany_filerName_	SubOffering_offeringShares_
SubCompany_irsNumber_	SubOffering_securityName_
SubContact_contactName_	SubOffering_typeOfPayment_
SubContact_contactPhoneNumber_	SubOffset_offsetAmount_
SubCoreg_fileNumber_	SubOffset_offsetFileNumber_
SubCoreg_filerCcc_	SubOffset_offsetFilingDate_
SubCoreg_filerId_	SubOffset_offsetForm_
SubCoreg_formType_	SubOffset_offsetId_
SubDate_effectivenessDateOfReport_	SubPayor_payorCcc_
SubDocument_conformedDocumentType_	SubPayor_payorId_
SubDocument_conformedName_	SubRef429_fileNumber_
SubDocument_description_	SubSalesShares_redeemedValue_
SubFee_feeAmount_	SubSalesShares_salesProceeds_
SubFee_feeBasis_	SubSalesShares_series_
SubFee_paymentMethod_	SubSerial_companyName_
SubFiler_fileNumber_	SubSro_sroId_
SubFiler_filerCcc_	SubTable_act_
SubFiler_filerId_	SubTable_depositorCik_
SubFiler_filerRelationship_	SubTable_depositorFileNumber_
SubFiler_formType_	SubTable_depository_
SubFiler_shellCompanyFlag_	SubTable_feeTableInProspectusFlag_
SubFiler_voluntaryFilerFlag_	SubTable_firstFilingIssuingEntityFlag_
SubFiler_wellKnownSeasonedIssuerFlag_	SubTable_fiscalYear_
SubFlag_confirmingCopyFlag_	SubTable_itemSubmissionType_
SubFlag_delayingAmendmentFlag_	SubTable_live_
SubFlag_overrideInternetFlag_	SubTable_periodOfReport_
SubFlag_returnCopyFlag_	SubTable_reference462b_
SubFlag_serialFlag_	SubTable_rule163Flag_
SubGlobal_enclosedFileCount_	SubTable_rule433Flag_
SubGroup_groupName_	SubTable_submissionType_
SubItem_itemId_	

5.5 Submission Data Nodes

All data related to the submission is stored in five types of XFDL nodes. These five node types are:

Node Type	Description
<field>	Text Field in the Viewer and/or the EDGARLink template
<check>	Checkbox in the Viewer and/or the EDGARLink template
<radio>	Radio Button in the Viewer and/or the EDGARLink template
<popup>	Popup in the Viewer and/or the EDGARLink template
<combobox>	Combo Box in the Viewer and/or the EDGARLink template

The rules for specifying one of these nodes are:

- EDGAR parsing is case-sensitive. You must use the *sid* names and valid values exactly as they are prescribed in the appendices and the attached master templates when constructing these nodes.
- A node must have a unique *sid* and a <value>.
<combobox sid="SubTable_submissionType_"><value>S-1</value></combobox>
- The <value> tag is optional if a field has no value. However, we suggest including the <value></value> tag for consistency.
- A <value> tag may only contain a single data item.
- The *sid* for these nodes must be of the form:

className_tagName_recordNumber

The className and tagName portions must exactly match the names provided in the Appendices and the underscores must always be present. Append the recordNumber to a *sid* only when you need to report multiple values for a particular category of data, e.g., Self Regulatory Organization (SRO) values. The recordNumber is simply an increasing integer number used to uniquely identify the multiple values for a particular field. The first instance of any field does not get a recordNumber; zero is implied. The second instance gets a recordNumber of 1, the third instance a recordNumber of 2, and so on. A couple of examples will make this clear.

Example 1:

For a field that has only a single value, such as the Submission Type combobox, the *sid* for the node will not have a recordNumber component. For this example, the node *sid* (field name) will be:

SubTable_submissionType_ (the trailing underscore must be present)

Example 2:

To report multiple values for a particular field, use the recordNumber to differentiate these values. For example, to report three SRO values, you would create these three node *sids*:

```
SubSro_sroId_  
SubSro_sroId_1  
SubSro_sroId_2
```

- f. White space between nodes is ignored. These two specifications are equivalent.

```
<field sid="SubTable_submissionType_"><value>S-1</value></field>  
<field sid="SubTable_submissionType_">  
  <value>S-1</value>  
</field>
```

For human readability of the XFDL, the latter case is preferred.

- g. The order of the nodes within a page only matters in two cases.

For enclosed documents in Templates 1, 2, 3, 4, and 6, SubDocument_conformedName_ must precede the document's <data> node.

For enclosed documents in Template 5,

SubModSeg_ModSegFileName_ must precede the document's <data> node.

- h. The ampersand (&) and left angle bracket (<) are not permitted in character data content of XML elements. If there is a need to include these characters in character data, such as a value, you must enclose the character data in an XML CDATA section. For example, a value for a form name which contains an embedded ampersand would need to be expressed as:

```
<value><![CDATA[EX-99.2I O&D BENEFTS]]></value>
```

5.6 Field Nodes

The only information required for a <field> node is a valid name and <value>. Appendix II lists all valid text field names.

```
<field sid="SubFiler_filerId_">  
  <value>0000000129</value>  
</field>
```

5.7 Checkbox Nodes

The only information required for a <check> node is a valid name and a <value>. Appendix II lists all valid checkbox names. It is recommended that you always include a <value> tag for consistency. This prevents a scenario where a checkbox is mandatory but the user provides no value. The only valid values for a <check> node are "on" and "off".

```
<check sid="SubFlag_confirmingCopyFlag_">  
  <value>on</value>  
</check>
```

5.8 Radio Button Nodes

The minimum information required for a <radio> node is a valid name and <value>. The only radio button in any of the templates is used to indicate if the submission is LIVE or TEST. A value of “on” means it is a LIVE submission. A value of “off” means it is a TEST submission. It is recommended that you always include a <value> tag for consistency. The only valid values for a <radio> node are “on” and “off”.

```
<radio sid=" SubTable_live_">
  <value>on</value>
</radio>
```

5.9 Popup Nodes

The only information required for a <popup> node is a valid name and <value>. Appendices IX–XIII list the valid values for all <popup> nodes.

```
<popup sid="SubFiler_formType_">
  <value>S-1</value>
</popup>
```

5.10 Combobox Nodes

The only information required for a <combobox> node is a valid name and <value>. Appendices IX–XIII list the valid values for all <combobox> nodes.

```
<combobox sid="SubTable_submissionType_">
  <value>S-4EF/A</value>
</combobox>
```

5.11 Data Nodes – (Enclosed Documents)

Each document enclosed on Page 2 of the submission must have a <data> node of the following format:

```
<data sid="data1">
  <filename>cover.txt</filename>
  <mimedata>
    VGhpcyBpcyBteSBjb3ZlciBsZXR0ZXIuDQpJdCBpcyBmb3IgdYS
    B0ZXN0IG9mIGegY29ycmVzcG9uZGVuY2Ugc3VibWlzc2lvbi4N
    Cg0KSSBob3BlIHRoaXMgd29ya3Mu
  </mimedata>
</data>
```

You may supply whatever you want as the *sid* for a data node, just as long as it is a valid XML name and it is unique to PAGE2 of the submission.

The <filename> is really the key item when defining a data node. It is the link to the SubDocument_conformedName_ or SubModSeg_ModSegFileName_ field that corresponds to this data node.

The <mimedata> block contains the MIME encoded document. A document must be encoded before you put it into the submission file. The MIME algorithm used by the PureEdge product is not proprietary, so any standard 64-character set based MIME encoding algorithm can be used.

Section 8.7, Sample CORRESP from Subtemplate4.xfd, provides a sample CORRESP submission with one enclosed document. Section 8.9, Sample BULK from Subtemplate6.xfd, provides a sample BULK submission with three enclosed documents. In the case of a bulk submission using Template6, each enclosed MIME document is a complete XFDL submission file.

6. VERIFYING FILER-CONSTRUCTED STRUCTURE

To verify that a filer-constructed submission file does not contain any structural errors, you can simply take advantage of the fact that the file is really XML. To verify the content:

- a. Rename the file, giving it an .xml suffix.
- b. Open the file in your browser.
- c. If the file contains structural errors, the browser will give you the offending line numbers.

Other commercially-available XML development tools can be used to identify syntax errors in your submission file.

7. HOW TO USE THE APPENDICES

This specification contains several appendices that define what is required for each electronic SEC form.

- a. Appendix I is a mapping of all EDGAR form types to the EDGARLink submission templates.
- b. Appendix II is a complete list of all the unique *sids* used in the six submission templates. This list also provides, for each *sid*, the node type (field, popup, combobox, check, or radio), a mapping of the node to the template(s) that use it, and whether the node is repeatable.

NOTE: You should never have a *sid* that is not in this list in a filer-constructed submission file.

- c. Appendices III through VIII show which PAGE1 header fields, or *sids*, are applicable to each form type for Templates 1 through 6.

Appendix III	Template1 form-type field dependencies
Appendix IV	Template2 form-type field dependencies
Appendix V	Template3 form-type field dependencies
Appendix VI	Template4 form-type field dependencies
Appendix VII	Template5 form-type field dependencies
Appendix VIII	Template6 form-type field dependencies

The following table explains the numeric codes that appear in the matrices in these six appendices.

Code	Status
1	Required
2	Conditionally Required (if any field in group has a value, all fields in the group must have a value)
3	When specifying a SubOffering entry, both the securityName and aggregatePrice are Required. The offeringShares and offeringAmountPerShare fields are optional.
4	Optional
5	Required for a confirming copy submission (i.e., for electronic submissions which are copies of official filings previously made in paper under a hardship exemption)
6	Required for all serial companies

The fields in Appendix II that have one of these numbers assigned are not included in Appendices III through VIII. These appendices indicate which of a template's conditional fields are applicable to each form type handled by the template.

To determine the complete set of valid *sids* for a given form type, you must refer to both Appendix II as well as one of these template-specific Appendices.

- d. Appendices IX through XIII show the valid values for the <field>, <combobox>, and <popup> nodes that are limited to a set of prescribed values.

Appendix IX	Template1 choice lists
Appendix X	Template2 choice lists
Appendix XI	Template3 choice lists
Appendix XII	Template4 choice lists
Appendix XIII	Template5 choice lists

8. SAMPLE FORMS

The sample filer-constructed submissions in this section demonstrate submissions constructed following the guidelines in this specification. There is at least one example for each of the six submission templates. All data in these examples is fictitious. These examples only include *sids* which are applicable to the given submission type.

8.1 Sample 24F-2NT from Subtemplate1.xfd

This is an example of a fee-bearing submission. The example excludes pages with no data. PAGE3, PAGE4, PAGE5, and PAGE6 are entirely absent.

```
<?xml version="1.0"?>
<XFDL version="5.0.0">
<page sid="PAGE1">
  <combobox sid="SubTable_submissionType_"><value>24F-2NT</value></combobox>
  <check sid="SubFlag_confirmingCopyFlag_"><value>on</value></check>
  <radio sid="SubTable_live_"><value>on</value></radio>
  <check sid="SubFlag_returnCopyFlag_"><value>off</value></check>
  <check sid="SubFlag_delayingAmendmentFlag_"><value>off</value></check>
  <field sid="SubFiler_filerId_"><value>01010101</value></field>
  <field sid="SubFiler_filerCcc_"><value>a#ofcorn</value></field>
  <field sid="SubFiler_fileNumber_"><value>033-04577</value></field>
  <field sid="SubContact_contactName_"><value>Joe Filer</value></field>
  <field sid="SubContact_contactPhoneNumber_"><value>br549</value></field>
  <popup sid="SubSro_sroId_"><value>AMEX</value></popup>
  <field sid="SubTable_periodOfReport_"><value>11/8/2000</value></field>
</page>
<page sid="PAGE2">
  <field sid="SubDocument_conformedName_"><value>primary.txt</value></field>
  <combobox sid="SubDocument_conformedDocumentType_">
    <value>24F-2NT</value></combobox>
  <field sid="SubDocument_description_"><value>primary document</value></field>
  <data sid="data1">
    <filename>primary.txt</filename>
    <mimedata>
      U1BFQ01NRU4gLSBUSElTIElTIEEgVEVTVCAtIE5PVCBBIExFR0
      FMIEZJTElORyANCg0KKioqKioqKioqKioqKioqKioqKioqKioq
      KioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKi
      oqKioqKioqDQo=
    </mimedata>
  </data>
</page>
<page sid="PAGE7">
  <field sid="SubPayor_payorId_"><value>01010101</value></field>
  <field sid="SubPayor_payorCcc_"><value>a#ofcorn</value></field>
  <combobox sid="SubFee_paymentMethod_"><value>LOCKBOX</value></combobox>
  <field sid="SubFee_feeAmount_"><value>7.58</value></field>
  <field sid="SubSalesShares_salesProceeds_"><value>100000</value></field>
  <field sid="SubSalesShares_redeemedValue_"><value>7500</value></field>
</page>
</XFDL>
```

8.2 Sample S-1 from Subtemplate1.xfd

This is another example of a fee-bearing submission. Note that it also excludes pages with no data.

```
<?xml version="1.0"?>
<XFDL version="5.0.0">
<page sid="PAGE1">
  <combobox sid="SubTable_submissionType_"><value>S-1</value></combobox>
  <check sid="SubFlag_confirmingCopyFlag_"><value>off</value></check>
  <radio sid="SubTable_live_"><value>on</value></radio>
  <check sid="SubFlag_returnCopyFlag_"><value>off</value></check>
  <check sid="SubFlag_delayingAmendmentFlag_"><value>off</value></check>
  <field sid="SubFiler_filerId_"><value>0101010101</value></field>
  <field sid="SubFiler_filerCcc_"><value>a#ofcorn</value></field>
  <field sid="SubContact_contactName_"><value>Joe Filer</value></field>
  <field sid="SubContact_contactPhoneNumber_"><value>br549</value></field>
  <popup sid="SubSro_sroId_"><value>AMEX</value></popup>
</page>
<page sid="PAGE2">
  <field sid="SubDocument_conformedName_"><value>primary.txt</value></field>
  <combobox sid="SubDocument_conformedDocumentType_">
    <value>S-1</value></combobox>
  <field sid="SubDocument_description_"><value>primary document</value></field>
  <data sid="data1">
    <filename>primary.txt</filename>
    <mimedata>
      U1BFQ01NRU4gLSBUSE1TIE1TIEEgVEVTVCAtIE5PVCBBIExFR0
      FMIEZJTE1ORyANCg0KKioqKioqKioqKioqKioqKioqKioqKioqKioq
      KioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKi
      oqKioqKioqDQo=
    </mimedata>
  </data>
</page>
<page sid="PAGE6">
  <field sid="SubPayor_payorId_"><value>0101010101</value></field>
  <field sid="SubPayor_payorCcc_"><value> a#ofcorn </value></field>
  <combobox sid="SubFee_paymentMethod_"><value>LOCKBOX</value></combobox>
  <field sid="SubFee_feeAmount_"><value>6.56</value></field>
  <combobox sid="SubOffering_securityName_"><value>Equity</value></combobox>
  <field sid="SubOffering_offeringShares_"><value>2000</value></field>
  <field sid="SubOffering_offeringAmountPerShare_"><value>40.00</value></field>
  <field sid="SubOffering_aggregatePrice_"><value>80000.00</value></field>
</page>
</XFDL>
```

8.3 Sample N-1 from Subtemplate1.xfd

This is an example of a non-fee-bearing submission.

```
<?xml version="1.0"?>
<XFDL version="5.0.0">
<page sid="PAGE1">
  <combobox sid="SubTable_submissionType_"><value>N-1</value></combobox>
  <check sid="SubFlag_confirmingCopyFlag_"><value>on</value></check>
  <radio sid="SubTable_live_"><value>on</value></radio>
  <check sid="SubFlag_returnCopyFlag_"><value>off</value></check>
  <check sid="SubFlag_delayingAmendmentFlag_"><value>on</value></check>
  <field sid="SubFiler_filerId_"><value>01010101</value></field>
  <field sid="SubFiler_filerCcc_"><value>a#ofcorn</value></field>
  <field sid="SubFiler_fileNumber_"><value>333-30266</value></field>
  <field sid="SubCoreg_filerId_"><value>20202020</value></field>
  <field sid="SubCoreg_filerCcc_"><value>a#ofbeef</value></field>
  <field sid="SubCoreg_fileNumber_"><value>033-04577</value></field>
  <field sid="SubContact_contactName_"><value>JoeFiler</value></field>
  <field sid="SubContact_contactPhoneNumber_"><value>123-456-7890</value></field>
  <combobox sid="SubTable_act_"><value>33</value></combobox>
  <field sid="SubRef429_fileNumber_"><value>333-30266</value></field>
  <popup sid="SubSro_sroId_"><value>AMEX</value></popup>
</page>
<page sid="PAGE2">
  <field sid="SubDocument_conformedName_"><value>doc01.txt</value></field>
  <combobox sid="SubDocument_conformedDocumentType_"><value>N-1</value></combobox>
  <field sid="SubDocument_description_"><value>doc1</value></field>
  <data sid="data1">
    <filename>doc01.txt</filename>
    <mimedata>
      U1BFQ01NRU4gLSBUSE1TIE1TIEEgVEVTVCAtIE5PVCBBIExFR0
      FMIEZJTE1ORyANCg0KKioqKioqKioqKioqKioqKioqKioqKioqKioq
      KioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKi
      oqKioqKioqDQo=
    </mimedata>
  </data>
  <field sid="SubDocument_conformedName_1"><value>cover01.txt</value></field>
  <combobox sid="SubDocument_conformedDocumentType_1"><value>COVER</value></combobox>
  <field sid="SubDocument_description_1"><value>doc2</value></field>
  <data sid="data2">
    <filename>cover01.txt</filename>
    <mimedata>
      U1BFQ01NRU4gLSBUSE1TIE1TIEEgVEVTVCAtIE5PVCBBIExFR0
      FMIEZJTE1ORyANCg0KKioqKioqKioqKioqKioqKioqKioqKioqKioq
      KioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKi
      oqKioqKioqDQo=
    </mimedata>
  </data>
</page>
<page sid="PAGE3">
  <field sid="SubInternet_internetAddress_"><value>joefiler@mailme.com</value></field>
</page>
</XFDL>
```


8.6 Sample 10-12B from Subtemplate3.xfd

This example shows two SEGMENT and one MODULE references on PAGE4. Notice that the CIK and CCC fields should not be included for a SEGMENT reference.

```
<?xml version="1.0"?>
<XFDL version="5.0.0">
<page sid="PAGE1">
  <combobox sid="SubTable_submissionType_"><value>10-12B</value></combobox>
  <check sid="SubFlag_confirmingCopyFlag_"><value>off</value></check>
  <radio sid="SubTable_live_"><value>on</value></radio>
  <check sid="SubFlag_returnCopyFlag_"><value>off</value></check>
  <field sid="SubFiler_filerId_"><value>20202020</value></field>
  <field sid="SubFiler_filerCcc_"><value>a#ofbeef</value></field>
  <field sid="SubCoreg_filerId_"><value>01010101</value></field>
  <field sid="SubCoreg_filerCcc_"><value>a#ofcorn</value></field>
  <field sid="SubContact_contactName_"><value>Joe Filer</value></field>
  <field sid="SubContact_contactPhoneNumber_"><value>br549</value></field>
  <popup sid="SubSro_sroId_"><value>AMEX</value></popup>
</page>
<page sid="PAGE2">
  <field sid="SubDocument_conformedName_"><value>cover.txt</value></field>
  <combobox sid="SubDocument_conformedDocumentType_">
    <value>10-12B</value></combobox>
  <field sid="SubDocument_description_"><value>doc one</value></field>
  <data sid="data1">
    <filename>primary.htm</filename>
    <mimedata>
      U1BFQ01NRU4gLSBUSE1TIE1TIEEgVEVTVCAtIE5PVCBBIExFR0
      AgICAgICAgICAgICAgICAgIEZPUk0gWFhYWPhYWA0KDQoqKioq
      KioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKi
      oqKioqKioqDQo=
    </mimedata>
  </data>
</page>
<page sid="PAGE3">
  <field sid="SubInternet_internetAddress_"><value></value></field>
  <check sid="SubFlag_overrideInternetFlag_"><value>on</value></check>
</page>
<page sid="PAGE4">
  <popup sid="ModSegReferenceData_modOrSegFlag_"><value>S</value></popup>
  <field sid="ModSegReferenceData_nickName_"><value>SEG20202020-1</value></field>
  <combobox sid="ModSegReferenceData_documentType_"><value>EX-1</value></combobox>
  <popup sid="ModSegReferenceData_modOrSegFlag_1"><value>S</value></popup>
  <field sid="ModSegReferenceData_nickName_1"><value>SEG20202020-2</value></field>
  <combobox sid="ModSegReferenceData_documentType_1"><value>EX-2</value></combobox>
  <popup sid="ModSegReferenceData_modOrSegFlag_2"><value>M</value></popup>
  <field sid="ModSegReferenceData_nickName_2"><value>MOD01010101-1</value></field>
  <field sid="ModSegReferenceData_cik_2"><value>01010101</value></field>
  <field sid="ModSegReferenceData_ccc_2"><value>a#ofcorn</value></field>
  <combobox sid="ModSegReferenceData_documentType_2"><value>EX-3</value></combobox>
</page>
</XFDL>
```


ACRONYMS

API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BNF	Backus Naur Form
CCC	CIK Confirmation Code
CIK	Central Index Key
DOS	Disk Operating System
DTD	Document Type Definition
EDGAR	Electronic Data Gathering, Analysis, and Retrieval
GUI	Graphical User Interface
SEC	Securities and Exchange Commission
<i>sid</i>	Scope Identifier
SRO	Self Regulatory Organization
XFDL	Extensible Forms Description Language
XML	Extensible Markup Language