MIMESGML Working Group D. Stinchfield
INTERNET-DRAFT EBT, Inc.
<draft-ietf-mimesgml-exch-01.ps >
Expires August 19,1996 February 19, 1996

Using SGML Open Catalogs and MIME to Exchange SGML Documents

Status of this Memo

Abstract

This proposal describes how SGML Open catalogs and MIME mechanisms are used to exchange SGML documents on the World Wide Web, or via email. Using the extension mechanism provided by Technical Resolution 9401:1995 (TR9401) [10]- TR9401 contains the technical description of SGML Open catalogs - this proposal describes new catalog keywords required for SGML document interchange. In addition, Uniform Resource Locators (URL) [8] are used to allow greater flexibility in the addressing of storage objects. This flexibility is intended to allow addressing of storage objects encapsulated in MIME messages, or addressable via the World Wide Web. A MIME body part containing an SGML Open catalog is tagged with the content type “application/sgml-open-catalog” [5].

Revision History

Changed Application/Catalog to Application/SGML-Open-Catalog.

Changed the syntax of Notation and added more to the description.

Added ENCODING and SYSTEM keywords.

Removed the keywords: CHARSET, BASESET, and CAPACITY. Will assume that the public id's used to define them in the SGML declaration will be unique within the catalog.

Changed BASEURL to BASE and added to the definition so that BASE can now be either an absolute URL or an absolute filename. Also described is how multiple BASE catalog entries are used.

Added a better description of how system identifiers are to be handled.

TR9401:1995 is being referenced instead of TR9401:1994.

User-define keywords are no longer required to begin with "X-" but it is strongly recommended.

Added appendix D, public identifiers for Notations.

Added an attribute to Semantics called "title".

Added "Usage Guidelines" section.

Removed information that is repeated from TR9401. This includes keyword descriptions and parts of the grammar.

Replaced Catalogs with 'SGML Open Catalogs' in the title.

1. Introduction

This proposal describes how SGML Open catalogs and MIME mechanisms are used to exchange SGML documents on the World Wide Web, or via email. Using the extension mechanism provided by Technical Resolution 9401:1995 (TR9401) [10]- TR9401 contains the technical description of SGML Open catalogs - this proposal describes new catalog keywords required for SGML document interchange. In addition, Uniform Resource Locators (URL) [8] are used to allow greater flexibility in the addressing of storage objects. This flexibility is intended to allow addressing of storage objects encapsulated in MIME messages, or addressable via the World Wide Web. A MIME body part containing an SGML Open catalog is tagged with the content type “application/sgml-open-catalog” [5].

Some benefits to using SGML Open catalogs to interchange SGML documents are:

- a client only needs a catalog to begin processing, it simply fetches the components referenced in the catalog as they are needed;
- a client that understands catalogs has a way to fetch components of a document that it doesn’t already have;
- document components do not have to be modified in order to be referenced in a catalog;
- components of a document can be distributed across many servers;
- catalogs do not depend on MIME, therefore, they can be used in other packaging schemes;
- the impact on MIME is minimized;
- catalogs are an implemented, proven, and widely used technology;
- a document's system identifiers can be referenced in a catalog and subsequently resolved by a client.

1.1 Overview

TR9401 defined catalog-keywords identify SGML document components, such as an SGML Declaration, a DTD, or a document entity. Catalogs containing only TR9401 catalog-keywords are useful for sharing documents between applications on a single system. These same catalogs are less useful for sharing documents between applications on remote and heterogeneous systems. For example, a system identifier that describes an absolute path to an MS-DOS file is useful on a PC but is not likely to be very useful on a UNIX system. Using the extension mechanism of TR9401, this proposal defines new catalog-keywords needed to address this problem, and others that are encountered when attempting to interchanging SGML documents over the World Wide Web, or via email.

The new keywords described in this proposal include: NOTATION, BASE, SEMANTICS, OVERRIDE, SYSTEM, and ENCODING. NOTATION is used to describe a document's SGML NOTATION declaration [3][11]. OVERRIDE indicates the TR9401 processing mode[10] for resolving external identifiers. BASE defines an absolute URL and is used for resolving relative URLs found in the catalog. SEMANTICS is used to reference semantic processing information such as stylesheets. SYSTEM provides a mapping between system identifiers and is useful for exchanging documents with SGML systems that are not catalog aware. ENCODING describes the encoding of catalog entries; it is possible for each catalog entry to have its own encoding.

In addition to catalog keywords, this proposal describes a system-independent SOI to be either a URL, a relative URL, or a MIME Content-ID. The usefulness of URLs and relative URLs is evident from World Wide Web. A Content-ID SOI identifies a document component contained in a MIME body part [13]. Typically, a Content-ID SOI describes the location of a document component within the same multipart message as the catalog.

2. Catalog Description

A catalog contains zero or more catalog entries. Each entry consists of a keyword, zero or more keyword attributes, and, usually, a storage object identifier defined as a URL, a relative URL, or a Content-ID. Relative URLs are resolved using the value of the BASE catalog entry. When no BASE entry exists relative URLs are resolved with respect to the location of the catalog.

2.1 Resolving System Identifiers

An SGML system identifier contains system-specific information used for locating an entity: a filename is an example of system-specific information. The kinds of system identifiers supported by an SGML system depends on the capabilities of its entity manager. Usually, there are two entity managers involved in a document exchange - in this document one of the entity managers is referred to as the sender's entity manager, and the other is referred to as the receiver's entity manager. Typically, the capabilities of the sender's entity manager is different from that of the receiver's, and the sender is usually unaware of the receiver's capabilities. The OVERRIDE and SYSTEM keywords are used to solve this problem. (Note, sometimes, especially for legacy SGML systems, this problem can only be solved by rewriting the document's system identifiers. Algorithms for rewriting system identifiers are beyond the scope of this document.)

Setting the OVERRIDE keyword to “YES” directs the receiving system to use the catalog to re-map system identifers occurring in the document. For example, the following is an entity declaration with a system identifier of “defc1.sgm”:

```
<!ENTITY c1 SYSTEM  “defc1.sgm” >
```

When a server creates a multipart message containing entity “c1” and a catalog that references it, the catalog entry for “c1” would look like this:

```
OVERRIDE “YES”
ENTITY   “c1”  “Content-ID:<==toons==>“
```

A receiving SGML system that understands SGML Open catalogs, the extensions proposed in this document, and MIME can simply process the catalog. It is likely that multipart messages received over the World Wide Web will be processed this way. For email, the MUA will likely save the body part defined by “Content-ID:<==toons==>“ to a file, and a helper application will rewrite the catalog to reflect the new location of “c1”:

```
OVERRIDE “YES”
ENTITY   “c1”  “c:\tmp\blurt.it”
```

It is useful for the helper application to save the body part using the original SOI of “c1”. Doing so, for this example, would relieve the receiver’s SGML system from having to process the catalog. The SYSTEM keyword facilitates this. For example, the aforementioned catalog contained in the multipart message can be rewritten to look like this:

```
OVERRIDE “YES”
ENTITY   “c1”         “Content-ID:<==toons==>“
SYSTEM   “defc1.sgm”  “Content-ID:<==toons==>“
```

The contents of the entity named “c1” are found in “Content-ID:<==toons==>“. The original system identifier for “Content-ID:<==toons==>“ is “defc1.sgm”. The receiver’s helper application can use this information to save the entity using the original SOI, defc1.sgm, defined for it in the document.

2.2 Catalog Keywords

A catalog contains entries for SGML document components. The order of the entries is important for the OVERRIDE, ENCODING, and BASE keywords. All entries are optional. A catalog can contain multiple entries with the same keyword. The following keywords are defined in TR9401 [10]:

SGMLDECL - SGML declaration
DOCUMENT - SGML document entity

```
DOCTYPE  - Document type declaration (DTD)
LINKTYPE - link type name
PUBLIC   - public external identifier
DTDDECL  - SGML declaration plus public identifier meant to
           match a public identifier given as part of the
           doctype declaration to reference the external subset.
ENTITY   - entity name
```

The following keywords are defined in this document using the grammar notation of TR9401:

```
NOTATION - notation name
SEMANTICS - name and type of the semantic information
BASE     - base URL
OVERRIDE - defines which TR9401 processing mode to use
ENCODING - character encoding
SYSTEM   - maps a system identifier from a document to a system
           identifier in the catalog
X-       - user-define keyword prefix
```

2.2.1 NOTATION

The NOTATION catalog keyword refers to data content notations defined or referenced in a document. The syntax for NOTATION is:

```
notation = ("NOTATION",  ps+,
             notation_name,
             (ps+, storage_object_identifier)?
            )
```

The storage object identifier is optional for NOTATION. The following example illustrates how NOTATION could be used for Java scripts:

```
<!NOTATION JuggleBalls SYSTEM "java/juggleballs.java">
<!ELEMENT Juggler - O CDATA>
<!ATTLIST Juggler notation NOTATION JuggleBalls>

<Juggler notation=JuggleBalls>

        input to JuggleBalls script, for example: specify number
        of items and juggling style.
```

A catalog entry for the NOTATION declaration described above would look like:

```
NOTATION  "JuggleBalls"  "http://www.bill.com/juggleballs.java"
```

The processing of notations is system dependent- there’s no way for a server to guarantee that a client can process a specific notation. The NOTATION keyword in the catalog may only give a hint, possibly a pointer, to a notation processor. It can be dangerous for the client to resolve a reference to a notation processor- loading and running a notation processor received from a remote, and potentially unsecured, site is dangerous. However, in a secure environment, exchanging scripts may be perfectly safe. See appendix D for some examples of some notations.

2.2.2 SEMANTICS

There may be semantic information such as stylesheets associated with a document. Semantic information is not required for parsing the document and can be ignored by the client. However, it is often desirable for a client to access appropriate semantic specifications. The syntax for the SEMANTICS keyword is:

```
semantics = ("SEMANTICS", ps+,
              semantic name, ps+,
              semantic type, ps+,
              semantic title, ps+,
              storage object identifier)
```

Here are two examples:

```
SEMANTICS "large-print"  "DSSSL"   "Wicked Large Print"
   "http://www.bill.com/style/large.sty"

SEMANTICS "toc"  "DSSSL"   "Table of Contents"  "toc.sty"
```

2.2.3 BASE

Relative URLs [4] are allowed in SOIs. Relative URLs can be resolved using a BASE keyword catalog entry. There can be more than one BASE keyword in a catalog. Relative URLs are resolved with respect to the closest previously specified BASE keyword, an example follows the syntax definition. If no BASE entry applies to a catalog entry, then the URL of the catalog is used for relative URL resolution. The syntax for the BASE keyword is:

```
base = ("BASE", ps+, absoluteURL)
```

Here’s an example of how BASE is used:

```
ENTITY   "Legal"      "legal.sgm"
BASE     "http://www.bill.com/docs/memo/mine/"
ENTITY   "MyEnding"   " ending.sgml"
```

The entity “Legal” is resolved relative to the URL of the catalog. The absolute URL for entity “MyEnding” is “http://www.bill.com/docs/memo/mine/ending.sgml”.

2.2.4 OVERRIDE

The OVERRIDE keyword defines which TR9401 processing mode the SGML system’s entity manager will use to resolve external identifiers. When OVERRIDE is “YES” the entity manager uses the catalog to resolve external identifiers, whether or not there is a system identifier defined for it in the document. When OVERRIDE is “NO” the entity manager uses the system identifiers found in the document when resolving references to external identifiers. The value of OVERRIDE is “NO” at the beginning of the catalog.

There can be more than one OVERRIDE keyword in a catalog. The OVERRIDE value that applies to an entry is the closest previously specified one. If the value for an OVERRIDE entry is omitted then “YES” is assumed.

The syntax for the OVERRIDE keyword is:

```
override = ("OVERRIDE”, (ps+, mode)?)

mode  =    (LIT, “YES”,LIT)  |
           (LITA,”YES”,LITA) |
           (LIT, “NO”, LIT)  |
           (LITA,”NO”, LITA)
```

Here's an example:

```
ENTITY   "Legal"      "legal.sgm"
OVERRIDE
ENTITY   "MyEnding"   " ending.sgml"
```

For this example override is “NO” for the entity “Legal” and “YES” for the entity “MyEnding”.

2.2.5 ENCODING

The ENCODING keyword provides a way to include entities in different encodings within a single document. The syntax of the ENCODING keyword is:

```
encoding = ( "ENCODING", ( ps+, encode_spec )? )
```

The ENCODING keyword indicates the encoding of catalog entries that follow. There can be more than one ENCODING entry in a catalog. When an ENCODING entry is found it supersedes the value of any preceding ENCODING entry.

ISO-646 [17] is the default when no ENCODING keyword is specified. An ENCODING entry with no argument resets the encoding to the default, ISO-646. ISO-646 was selected as the default value because of the special role it plays in SGML: ISO-646 is the syntax-reference character set of the reference concrete syntax [3], pg 476.

For example, the following catalog describes catalog entries, each with a different encoding:

```
ENCODING "SHIFT-JIS"
DOCUMENT  "http://www.goeast.com/anaxi.sgm"
ENCODING "ISO-10646-UTF7"
DOCTYPE  "MEMO"       "http://www.gowest.com/dtds/memo.dtd"
ENCODING "SHIFT-JIS"
ENTITY   "MyEnding"   "http://www.goeast.com/ending.sgml"
ENTITY   "Legal"      "http://www.goeast.com/company/legal.sgm"
```

The document entity and the entities MyEnding and Legal are encoded in SHIFT-JIS while the DTD is encoded in ISO-10646-UTF7.

2.2.6 SYSTEM

The SYSTEM keyword is used to map an original system identifier found in a document to the SOI used for it in the catalog. The SYSTEM keyword is very similar to the SYSTEM keyword used by nsgmls [16]. The syntax for the SYSTEM keyword is:

```
        system  = (“SYSTEM”,
                    original_soi,
                    effective_soi
                   )
```

For an example of how SYSTEM is used refer to the previous section entitled “Resolving System Identifiers”.

2.2.7 User-Defined Keywords

It is strongly recommended that user-defined keywords begin with “X-”. This allows the catalog-parser to easily determine if a keyword is a user-defined keyword.

2.3 Storage Object Identifiers

Three types of SOIs are defined in this section. The first defines an SOI in terms of URLs. The second defines an SOI in terms of a MIME Content-ID. And the last defines an SOI using TR9401's definition of an SOI.

The syntax for an SOI is:

```
     storage object identifier =
            url object identifier             |
            content id object identifier      |
            TR9401 storage object identifier
```

2.3.1 URLs as SOIs

An SOI can be a URL [8] or a relative URL [4]. The SOI will need to be parsed to determine the SOI type.

2.3.2 The Content-ID SOI

A Content-ID SOI specifies a MIME message body part. Note, the Content-ID SOI is expected to be replaced in the future with the cid URL [8]. The syntax for Content-ID based SOI is:

```
  content id object identifier =
           "Content-ID" ":" msg-id ; as defined in RFC 1521 [13]
  msg-id      =  as defined in RFC 822 [15]
```

3. Catalog Syntax

```
catalog =
   ( ps*, ( (catalog_entry | user_defined), ps+ )* )

catalog_entry  =
          TR9401:1995_keywords |
          notation             |
          semantics            |
          base                 |
          override             |
          system               |
          encoding

TR9401:1995_keywords = refer to TR9401 [10] for a description of
                       keywords for sgmldecl, document, doctype,
                       public, entity, and linktype
```

```
notation = ("NOTATION",  ps+,
             notation_name,
             (ps+, storage_object_identifier)?
            )

notation_name = entity_name_spec

entity_name_spec  = as defined in TR9401 [10]

semantics = ("SEMANTICS", ps+,
              semantic_name, ps+,
              semantic_type, ps+,
              semantic_title, ps+,
              storage_object_identifier)

semantic_name = entity_name_spec

semantic_type = entity_name_spec

semantic_title = entity_name_spec

base = ("BASE",  ps+, URL_spec)

URL_spec = (LIT,  absoluteURL, LIT)  |
           (LITA, absoluteURL, LITA)

absoluteURL = see "absoluteURL" in RFC 1808 [4]

override = ("OVERRIDE", (ps+, mode)?)

mode  = (LIT, "YES",LIT)  |
        (LITA, "YES",LITA) |
        (LIT, "NO", LIT)  |
        (LITA, "NO", LITA)

encoding = ( "ENCODING" ( ps+, encode_spec )? )

encode_spec = entity_name_spec

system  = ("SYSTEM", ps+,
            original_soi, ps+,
            effective_soi
           )

original_soi  = storage_object_identifier

effective_soi = storage_object_identifier

user_defined = ("X-", keyword)
```

```
storage_object_identifier =
          url_object_identifier             |
          content_id_object_identifier      |
          TR9401_storage_object_identifier

url_object_identifier =
          as defined in RFC 1738[8]

content_id_object_identifier =
              "Content-ID" ":" msg-id

msg-id      =  as defined in RFC 822 [15]

TR9401_storage_object_identifier =
          "storage object identifier" as defined in TR9401 [10]

keyword  =  as defined by TR9401

ps = as defined in TR9401 [10]
```

4. Usage Guidelines

There are some ambiguities in TR9401 which can be easily avoided by adhering to these guidelines:

quote all keyword-attributes using either single or double quotes. This allows the parser to determine what is an attribute, and what is not, following a user-defined keyword;

surround comments with whitespace. In other words, don’t start a comment just after a token, doing so can lead to parsing ambiguities.

5. Examples

The SGML document used in all the examples is composed of the following components:

- o an SGML declaration, defined in Appendix A;
- o a Document type declaration (DTD), defined in Appendix B;
- o an SGML document entity, defined in Appendix C;
- o two SGML entities, defined in Appendix C;
- o a figure entity, not defined in this draft.

In all examples the components of this SGML document are spread across multiple servers, except for the example entitled "Sending a Catalog and All of its Components”.

Each example defines its own unique catalog. The catalog varies from example to example depending on the number of document components sent along with it. The sender decides how many components to include in the MIME message.

A document component that is not included in the MIME message can be resolved in one of two ways: 1) the client requests the component from its cache, that is, the component had been fetched while processing a previous request, or 2) the client requests the component using SOI defined for it in the catalog.

The definitions for the following external identifiers are not included in this document:

formal public identifiers:

```
ISO 646:1983//CHARSET
    International Reference Version (IRV)//ESC 2/5 4/0
ISO 8879-1986//ENTITIES Numeric and Special Graphic//EN
```

system identifier:

```
../style/all.sty" - DSSSL style sheet
```

5.1 Sending Only A Catalog

In this example only the catalog is sent to the client. When the client’s SGML system is capable of handling SGML Open catalogs, along with the extensions proposed in this document, the catalog can be passed, without modification, to the client’s SGML System. Some pre-processing may be required when the client’s SGML system cannot handle these kinds of catalogs. For example, a catalog-aware SGML system that does not understand the keywords proposed in this documents or URLs would need pre-processor to do the following:

1. fetch all of the components referenced in the catalog,

2. store the components locally, and
3. update the catalog to reflect the new locations.

Here are some reasons for sending just the catalog:

- o the server might only store catalogs, that is, the server does not store any document components;
- o the client may have requested only the catalog. Perhaps the client wants to compare the contents of this catalog with the contents of a different catalog. Or maybe the client already has most, if not all, of the document's components cached;
- o the server may want to keep network traffic down by increasing the likelihood that the client will get a cache hit on catalog entries.

5.1.1 MIME Message Content

```
MIME-Version: 1.0
Content-Type: Application/SGML-Catalog; charset=us-ascii

SGMLDECL  "http://www.ebt.com/decl/ebtsgml.dcl"
OVERRIDE  "YES"
PUBLIC    "ISO 646:1983//CHARSET
          International Reference Version (IRV)//ESC 2/5 4/0"
          "http://www.iso.ch/charset/6461983.cha"
PUBLIC    "ISO Registration Number 100//CHARSET
          ECMA-94 Right-hand Part of Latin Alphabet Nr.1//ESC 2/13 4/1"
          "http://www.iso.ch/charset/ecma94.cha"
PUBLIC    "-//EBT//CAPACITY CoolCaps 1.0//"
          "http://www.ebt.com/decl/coolcaps.cap"
PUBLIC    "-//EBT//SYNTAX  SinSyn 0.1//"
          "http://www.ebt.com/decl/syntax/sinsyn.syn"
BASE      "http://www.bill.com/docs/memo/mine/"
DOCUMENT  "anaxi.sgm"
DOCTYPE   "MEMO"        "../../dtds/memo.dtd"
PUBLIC    "ISO 8879-1986//ENTITIES Numeric and Special Graphic//EN"
          "http://www.wcs.com/usr/wcs/isonum.ent"
ENTITY    "%ISOnum"    "http://www.wcs.com/usr/wcs/isonum.ent"
ENTITY    "MyEnding"   "ending.sgml"
ENTITY    "Legal"      "../company/legal.sgm"
SEMANTICS "large-print"  "DSSSL"   "../style/all.sty"
```

5.2 Sending a Catalog and a Document Entity

This example describes how to send a catalog and a document entity using a Multipart message. This is a likely scenario for Web-based browsers where simultaneous rendering and resolution of external identifiers are necessary. For this example the server assumes the client already has the SGML declaration, the DTD, and a stylesheet - if this is not the, the client can easily request them. The document entity will likely contain enough text for a browser to render meaningful text on the display device, but it won't include the many entities that the text may link to. These external objects, like figures, can be resolved by either the entity manager, while the application is rendering the text, or on user demand, as for hyperlinked information.

5.2.1 MIME Message Content

```
MIME-Version: 1.0
Content-Type: Multipart/Mixed; boundary=let-go-of-my-leg;

--let-go-of-my-leg
Content-Type: Application/SGML-Catalog; charset=us-ascii

SGMLDECL  "http://www.ebt.com/decl/ebtsgml.dcl"
OVERRIDE  "YES"
PUBLIC    "ISO 646:1983//CHARSET
           International Reference Version (IRV)//ESC 2/5 4/0"
          "http://www.iso.ch/charset/6461983.cha"
PUBLIC    "ISO Registration Number 100//CHARSET
          ECMA-94 Right-hand Part of Latin Alphabet Nr.1//ESC 2/13 4/1"
         "http://www.iso.ch/charset/ecma94.cha"
PUBLIC    "-//EBT//CAPACITY CoolCaps 1.0//"
          "http://www.ebt.com/decl/coolcaps.cap"
PUBLIC    "-//EBT//SYNTAX  SinSyn 0.1//"
          "http://www.ebt.com/decl/syntax/sinsyn.syn"
BASE      "http://www.bill.com/docs/memo/mine/"
DOCUMENT  "Content-ID:<bookemdano@toons.com>"
DOCTYPE   "MEMO"        "../../dtds/memo.dtd"
PUBLIC    "ISO 8879-1986//ENTITIES Numeric and Special Graphic//EN"
          "http://www.wcs.com/usr/wcs/isonum.ent"
ENTITY    "%ISOnum"     "http://www.wcs.com/usr/wcs/isonum.ent"
ENTITY    "MyEnding"    "ending.sgml"
ENTITY    "Legal"       "../company/legal.sgm"
SEMANTICS "large-print"  "DSSSL"   "../style/all.sty"
SYSTEM    "anaix.sgm"  "Content-ID:<bookemdano@toons.com>"

--let-go-of-my-leg
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <bookemdano@toons.com>
Content-Disposition: attachment; filename="anaxi.sgm"

include document entity from Appendix C
--let-go-of-my-leg--
```

5.3 Sending a Catalog and All Document Components

Sending a catalog and all of a document’s components at once is done using a Multipart message.

5.3.1 MIME Message Content

```
MIME-Version: 1.0
Content-Type: Multipart/Mixed; boundary=go-speed-racer

--go-speed-racer
Content-Type: Application/SGML-Catalog; charset=us-ascii

SGMLDECL "Content-ID:<joejitsu@toons.com >"
OVERRIDE "YES"
PUBLIC    "ISO 646:1983//CHARSET
           International Reference Version (IRV)//ESC 2/5 4/0"
          "Content-ID:<mumbles@toons.com >"
PUBLIC    "ISO Registration Number 100//CHARSET
          ECMA-94 Right-hand Part of Latin Alphabet Nr.1//ESC 2/13 4/1"
          "Content-ID:<underdog@toons.com >"
PUBLIC    "-//EBT//CAPACITY CoolCaps 1.0//"
          "Content-ID:<mastercylinder@toons.com >"
PUBLIC    "-//EBT//SYNTAX  SinSyn 0.1//"
          "Content-ID:<lippythelion@toons.com >"
DOCUMENT  "Content-ID:<andhardyharhar@toons.com >"
DOCTYPE   "MEMO"       "Content-ID:<mahilagorilla@toons.com >"
PUBLIC    "ISO 8879-1986//ENTITIES Numeric and Special Graphic//EN"
          "Content-ID:<mrpeebles@toons.com >"
ENTITY    "%ISOnum"    "Content-ID=<wallygator@toons.com >"
ENTITY    "MyEnding"   "Content-ID:<mrwhoopy@toons.com >"
ENTITY    "Legal"      "Content-ID:<johnnyquest@toons.com >"
SEMANTICS "large-print" "DSSSL" "Content-ID:<hodgyandbandit@toons.com>"
SYSTEM    "anaix.sgm"  "Content-ID:<andhardyharhar@toons.com>"
SYSTEM   "/usr/wcs/dtd/memo.dtd" "Content-ID:<mahilogorilla@toons.com>"
SYSTEM   "/usr/des/ending.sgm"   "Content-ID:<mrwhoopy@toons.com>"

--go-speed-racer
Content-Type:Application/SGML; charset=us-ascii
Content-ID:"<joejitsu@toons.com>"

description of SGML declaration from Appendix A is included here

--go-speed-racer
Content-Type:Application/SGML; charset=us-ascii
Content-ID:"<mumbles@toons.com >"

ISO 646 character set definition included here
```

```
--go-speed-racer
Content-Type:Application/SGML; charset=us-ascii
Content-ID:"<mastercylinder@toons.com >"

description of Capacity from Appendix A is included here

--go-speed-racer
Content-Type:Application/SGML; charset=us-ascii
Content-ID:"<lippythelion@toons.com >"

description of Syntax from Appendix A is included here

--go-speed-racer
Content-Type:Application/SGML; charset=us-ascii
Content-ID:"<underdog@toons.com >"

Contents of ISO Registration Number 100//CHARSET ECMA-94 Right-hand
Part of Latin Alphabet Nr.1//ESC 2/13 4/1 included here

--go-speed-racer
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <andhardyharhar@toons.com >
Content-Disposition: attachment; filename="anaxi.sgm"

include Document entity as described in Appendix C

--go-speed-racer
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <mahilogorilla@toons.com >
Content-Disposition: attachment; filename="memo.dtd"

include DTD as described from Appendix B

--go-speed-racer
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <mrpeebles@toons.com >

ISO 8879-1986 Entity set included here

--go-speed-racer
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <wallygator@toons.com >

include entity set defined for %ISOnum

--go-speed-racer
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <mrwhoopy@toons.com >
Content-Disposition: attachment; filename="ending.sgm"

include entity MyEnding as described in Appendix C
```

```
--go-speed-racer
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <johnnyquest@toons.com >

include entity Legal as described in Appendix C

--go-speed-racer
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <hodgyandbandit@toons.com >

included here is a bunch of DSSSL-Lite

--go-speed-racer--
```

5.4 Sending a Catalog for a Non-Document Entity

This example describes what a server might send in response to a request for a non-document entity. All of the previous examples assume that the original request was for a document entity. SGML documents can get very deep and contain a large number of external identifier references. Likewise, the complete catalog for a document could get very large: a "complete catalog" contains all of the external identifiers referenced in all of a document's entities. There is no need to send the complete. All that's needed are enough entries in the catalog for the client system to resolve references declared in the entity being transferred. For example, Appendix C.2 defines an entity called "Legal" which includes a reference to an entity called "MyEnding". A request for "Legal" would result in a Multipart/Related message that looks like this:

```
MIME-Version: 1.0
Content-Type: Multipart/Mixed; boundary=let-go-of-my-leg

--let-go-of-my-leg
Content-Type: Application/SGML-Catalog; charset=us-ascii

BASE     "http://www.bill.com/docs/memo/mine/"
OVERRIDE "YES"
ENTITY   "Legal" "Content-ID:<bookemdano@toons.com >"
ENTITY   "MyEnding"   "ending.sgml"

--let-go-of-my-leg
Content-Type: Application/SGML; charset=us-ascii
Content-ID: <bookemdano@toons.com >

include entity "Legal" from Appendix C.2

--let-go-of-my-leg--
```

6. Security Considerations

SGML documents, like other compound documents, may contain entities whose media-types present security concerns, e.g. Application/PostScript. Further, SGML may contain explicit processing instructions for a presentation or composition system; use of such instructions present concerns similar to those of Application/PostScript.

The use of active media-types with Notation declarations can provide an opportunity for the sender to execute a script or other code on the recipient's machine.

7. Acknowledgments

Thanks to Andre Alguero, Jeff Cutler-Stamm, Steve DeRose, Chris Maden, Gavin Nicol, and Bill Smith for helping me with the content and structure of this document. Thanks to Martin Bryan, James Clark, John Klensin, and Ed Levinson for the many discussions and debates that helped me to clarify, I hope, many of the ideas contained in this document. Thanks also to Wayne Wohler of IBM for his help on SGML declarations.

8. References

[1] Wayne Wohler, "SGML declarations",
http://www.sil.org/sgml/wlw11.html
[2] Eric van Herwijnen, "Practical SGML", Second Edition,
Kluwer Academic Publishers, 1994, ISBN 0-7923-9434-8
[3] Charles F. Goldfarb, "The SGML Handbook",
Oxford University Press, 1994, ISBN 0-19-853737-9
[4] R. Fielding, "Relative Uniform Resource Locators", RFC 1808,
ftp://ds.internic.net/rfc/rfc1808.txt
[5] P. Grosso, “The Application/SGML-Open-Catalog Content Type”, RFC
[6] Daniel W. Connolly, HTML 2.0 SGML declaration found at
http://www.w3.org/hypertext/WWW/MarkUp/html-spec/html.decl
[7] T. Berners-Lee, "Universal Resource Identifiers in WWW:
A Unifying Syntax for the Expression of Names and Addresses of
Objects on the Network as used in the World-Wide Web", RFC 1630
[8] T. Berners-Lee, L. Masinter, and M. McCahill,
"Uniform Resource Locators (URL)", RFC 1738
[10] Paul Grosso, "Entity Management", SGML Open
Technical Resolution 9401:1995 (Amendment 1 to TR9401),
http://www.sgmlopen.org/sgml/docs/library/9401.htm
[11] Charles F. Goldfarb, "Entity Management in SGML", 11/30/93
[12] Sollins, K. and Masinter, L.,
"Functional Requirements for Uniform Resource Names", RFC 1737
[13] N.Borenstein, "MIME (Multipurpose Internet Mail Extensions)
Part One: Mechanisms for Specifying and Describing the Format of
Internet Message Bodies", RFC 1521
[14] "ISO 8879:1986 Information processing - The and office systems -
Standard Generalized Markup Language (SGML)",
Geneva, 15 October 1986
[15] D. H. Crocker, "Standard for the Format of ARPA Internet Text
Messages", RFC 822
[16] J. Clark “nsgmls- a validating sgml parser “,
http://www.jclark.com/nsgmls.txt

[17] ISO 646- ISO 7-bit coded character set for information interchange

9. Authors' Address

Don Stinchfield
Electronic Book Technologies, Inc.
One Richmond Square
Providence, RI 02906
(401) 421-9550 x280
des@ebt.com

10. Appendix A: SGML declaration Used In The Examples

This Appendix contains the definitions for the SGML declaration, for the CAPACITY parameter, and for the SYNTAX parameter. The SGML declaration is a modified version of the one used for HTML 2.0 [6] - I changed the CAPACITY and SYNTAX declarations so that they referenced public identifiers. The following external identifiers are reference in the SGML declaration:

- BASESET "ISO 646:1983//CHARSET
  International Reference Version (IRV)//ESC 2/5 4/0"
- BASESET "ISO Registration Number 100//CHARSET
  ECMA-94 Right-hand Part of Latin Alphabet Nr.1//ESC
  2/13 4/1"
- CAPACITY PUBLIC "-//EBT//CAPACITY CoolCaps 1.0//"
- SYNTAX PUBLIC "-//EBT//SYNTAX SinSyn 0.1//"

10.1 SGML declaration

```
<!SGML  "ISO 8879:1986"
--
      Copied this from the SGML declaration for HyperText Markup
Language (HTML).
--

CHARSET
         BASESET  "ISO 646-1983//CHARSET
                   International Reference Version
                   (IRV)//ESC 2/5 4/0"
         DESCSET  0   9   UNUSED
                  9   2   9
                  11  2   UNUSED
                  13  1   13
                  14  18  UNUSED
                  32  95  32
                  127 1   UNUSED
         BASESET  "ISO Registration Number 100//CHARSET
                  ECMA-94 Right-hand Part of
                  Latin Alphabet Nr.1//ESC 2/13 4/1"

         DESCSET  128  32   UNUSED
                  160  96    160

CAPACITY   SGMLREF
           TOTALCAP        150000
           GRPCAP          150000
           ENTCAP          150000

SCOPE      DOCUMENT
SYNTAX
```

```
           SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
             17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127
           BASESET  "ISO 646-1983//CHARSET
                   International Reference Version
                   (IRV)//ESC 2/5 4/0"
           DESCSET  0 128 0
           FUNCTION
                   RE          13
                  RS           10
                  SPACE        32
                  TAB SEPCHAR   9


           NAMING   LCNMSTRT ""
                  UCNMSTRT ""
                  LCNMCHAR ".-"
                  UCNMCHAR ".-"
                  NAMECASE GENERAL YES
                           ENTITY  NO
           DELIM    GENERAL  SGMLREF
                  SHORTREF SGMLREF
           NAMES    SGMLREF
           QUANTITY SGMLREF
                  ATTSPLEN 2100
                  LITLEN   1024
                  NAMELEN  72    -- somewhat arbitrary; taken from
                                 internet line length conventions --
                  PILEN    1024
                  TAGLVL   100
                  TAGLEN   2100
                  GRPGTCNT 150
                  GRPCNT   64


FEATURES
  MINIMIZE
    DATATAG  NO
    OMITTAG  YES
    RANK     NO
    SHORTTAG YES
  LINK
    SIMPLE   NO
    IMPLICIT NO
    EXPLICIT NO
  OTHER
    CONCUR   NO
    SUBDOC   NO
    FORMAL   YES
  APPINFO    "SDA"  -- conforming SGML document Access application
                     --
>
```

11. Appendix B: DTD Used In The Examples

The DTD listed below is a modified version of the one found on page 33 of Eric van Herwijnen's book called "Practical SGML" [2]. The following external identifier is used in the DTD:

```
     <!ENTITY % ISOnum PUBLIC
                "ISO 8879-1986//
                ENTITIES Numeric and Special Graphic//EN"
                "/usr/wcs/isonum.ent" >
```

The above definition is for a parameter entity and it contains both a public identifier and a system identifier. The examples have both in the catalog.

11.1 DTD

```
<!--  DTD for simple office memoranda                              -->
<!ENTITY % doctype "MEMO"  -- document type generic identifier -->
<!--        ELEMENTS    MIN     CONTENT      (EXCEPTIONS)          -->
<!ELEMENT   %doctype;   - -    ((TO & FROM),  BODY, CLOSE?)         >
<!ELEMENT   TO          - O    (#PCDATA)                            >
<!ELEMENT   FROM        - O    (#PCDATA)                            >
<!ELEMENT   BODY        - O    (P)*                                 >
<!ELEMENT   P           - O    (#PCDATA | Q)*                       >
<!ELEMENT   PREF        - O    EMPTY                                >
<!ELEMENT   Q           - -    (#PCDATA)                            >
<!ELEMENT   CLOSE       - -    (#PCDATA)                            >
<!--        ELEMENTS    NAME   VALUE                DEFAULT        -->
<!ATTLIST   %doctype    STATUS (CONFIDEN|PUBLIC)    PUBLIC          >
<!ATTLIST   P           id     ID                   #IMPLIED        >
<!ATTLIST   PREF        refid  IDREF                #IMPLIED        >
<!ENTITY    %  ISOnum PUBLIC "ISO 8879-1986//ENTITIES Numeric and
                              Special Graphic//EN"
                              "/usr/wcs/isonum.ent">
%ISOnum;
```

12. Appendix C: SGML document Used In The Examples

The SGML document defined in this appendix is broken up into 3 parts: an SGML document entity and two SGML Entities. The SGML document entity contains references to external identifiers in the DOCTYPE and ENTITY declarations:

- This one contains both a public identifier and a system identifier:

```
<!DOCTYPE MEMO  PUBLIC "-//EBT//DTD Released Memo//EN"
                       "/usr/wcs/dtd/memo.dtd" [...] >
```

- This ENTITY declaration has system identifier and a system identifiers parameter:

```
<!ENTITY  MyEnding SYSTEM "/usr/des/ending.sgm"  >
```

- This one specifies a system identifier without specifying a system identifier parameter (this is provided for in the SGML Standard for implementers that want to resolve System Identifiers from the entity name alone [3, p378]):

```
<!ENTITY  Legal SYSTEM  >
```

12.1 SGML document entity

```
<!DOCTYPE MEMO  PUBLIC
     "-//EBT//DTD Released Memo//EN"
     "/usr/wcs/dtd/memo.dtd" [
<!ENTITY  Legal     SYSTEM  >
<!ENTITY  MyEnding  SYSTEM "/usr/des/ending.sgm"  >
] >
<MEMO>
<TO>Anaximander
<FROM>Cool Papa Shad
<BODY>
&Legal;
<P>Yo Anax, you’ve got a bizarre name!</P>
<P>&MyEnding;</P>
</BODY>
</CLOSE>
</MEMO>
```

12.2 Entity Named "Legal"

```
<P>If you or anyone you know tries to read this email then you’re in
really big trouble!</P>
<P>You know this is the end of the document when you see &MyEnding;</P>
```

12.3 Entity Named "MyEnding"

```
<Q>Regards, Don</Q>
```

13. Appendix D: NOTATIONS

13.1 Useful Notations

For Tcl I have taken the ISBN number from John K. Ousterhout's book "Tcl and the Tk Toolkit" to create a Formal Public Identifier:

```
<!NOTATION TCL  PUBLIC
 "ISBN 0-201-63337-X::Addison-Wesley//NOTATION Tcl and the Tk
Toolkit//EN"
```

13.2 Notations from the DocBook DTD

The following notations are defined in the DocBook DTD (the DocBook DTD can be found from http://www.ora.com/davenport/README.html):

```
<!NOTATION TEX PUBLIC
     "+//ISBN 0-201-13448-9::Knuth//NOTATION The TeXbook//EN"  >

<!NOTATION EPS PUBLIC
  "+//ISBN 0-201-18127-4::Adobe//NOTATION PostScript Language Ref.
Manual//EN" >

<!NOTATION FAX PUBLIC
"-//USA-DOD//NOTATION CCITT Group 4 Facsimile Type 1 Untiled
Raster//EN">

<!NOTATION IGES PUBLIC
"-//USA-DOD//NOTATION (ASME/ANSI Y14.26M-1987) Initial Graphics
Exchange Specification//EN" >
```