

FREQUENTLY ASKED QUESTIONS ABOUT THE EXTENSIBLE MARKUP LANGUAGE

THE XML FAQ



한글 번역판

VERSION 1.21 (1998년 2월 3일)

이 문서는 W3C(the World Wide Web Consortium)의 XML SIG(Special Interest Group)을 대신하여 전세계의 FAQ 독자들 뿐 아니라 **Peter Flynn** (University College Cork), **Terry Allen**, **Tom Borgman** (Harlequin Ltd), **Tim Bray** (Textuality, Inc), **Robin Cover** (Summer Institute of Linguistics), **Christopher Maden** (O'Reilly & Associates), **Eve Maler**(Arbortext, Inc), **Peter Murray-Rust** (Nottingham University), **Liam Quin**, **Michael Sperberg-McQueen**(University of Illinois at Chicago), **Joel Weber** (MIT), **Murata, Makoto** (Fuji Xerox Information Systems) 및 다른 많은 XML SIG 멤버들에 의해 유지되고 있다. 수정사항이나 추가사항등이 있을 경우 이 문서의 마지막에 있는 폼을 이용해 주기 바란다.

최근의 변화들

1998년 2월 3일

- 맥(Mac) 아이콘의 추가 (Martin Winter와 그외 다른분께 감사드립니다.)
- Draft를 참조문에서 스펙(spec)으로 이동
- 개정판 컬러의 변경
- RMD의 삭제
- 잘못된 URL들의 수정
- [1.21] URL들과 번역의 사소한 수정
- MIME 타입에 XUA 추가
- 오타 등 작은 수정

지난 버전 이후로 새롭게 추가된 문단은 문단 앞쪽에 "¶"로 표시하였으며, 변경된 문단은 "\$"로 표시하였다. 또한 앞으로는 없어지겠지만 현재는 유지되는 문단은 "⌘"로 표시하였다.

요약

이 문서는 XML에 관한 가장 자주 질문되는 문제들을 답변과 함께 싣고 있다. 이 문서의 목적은 이 문서를 개발자들, 관심있는 독자들에게 첫번째 자료가 되는 것이다. 하지만 이 문서는 "#FAQ-SPEC"-XML 스펙 일부분으로 간주할 수 없다.

문서의 구성

이 FAQ는 여러 부분으로 이루어져 있다. a) [URL](#) 변환, "#FAQ-USING"은 "#FAQ-AUTHOR"과 "#FAQ-DEVELOPER"은 각각의 섹션내의 질문들의 번호는 독립적으로 붙여진다. 따라서 버전에 따라 번호는 바뀔 수 있으며, 각 파트와 질문들의 번호 뿐 아니라 각종 표현들과 제안들은 버전 번호를 참조하여야 한다(위에 있는 "#FAQ-REVISION"-Revision History를 보라).

이 문서를 자체에 관한 버그 리포트, 개선을 위한 제안, 기타 의견 등이 있다면 이 문서의 끝에 있는 "#FAQ-FORM"-양식을 이용해 주시기 바란다. "#FAQ-SPEC"-XML Specification에 대한 의견은 "<http://www.w3.org/target='new'-W3C>"로 보내기 바란다.

문서의 이용

SGML 시스템에서 이용할 수 있는 "#FAQ-SGML"-SGML 파일은 "<http://www.ucc.ie/xml/faq.sgml>" target="new"-<http://www.ucc.ie/xml/faq.sgml>에서 구할 수 있다(또한 이 문서는 "<http://www.sq.com/products/panorama/panor-fe.htm>" target="new"-Panorama와 "<http://www.citec.fi/mdp/index.html>" target="new"-Citec의 SGML 브라우저들 통해 온라인으로 볼 수 있다; 로컬에서 이러한 문서를 볼 수 있도록 하기 위해서 "<http://www.ucc.ie/xml/xmlview.exe>" target="new"-XMLview 또는 "<http://www.ucc.ie/xml/catalog>" target="new"-Catalog set을 다운로드 받을 수도 있다).

같은 문서가 "<http://www.ucc.ie/xml/>" target="new"-<http://www.ucc.ie/xml/>에서 HTML 브라우저 (예를 들어 "<http://www.netscape.com/>" target="new"-Netscape Navigator, "<http://www.microsoft.com/>" target="new"-Microsoft Internet Explorer, "<http://www.spry.com/>" target="new"-Spry Mosaic, "<http://www.ncsa.edu/>" target="new"-NCSA Mosaic, "<http://kufacts.cc.ukans.edu/>" target="new"-Lynx, "<http://opera.nta.no/>" target="new"-Opera, "<http://www.cs.indiana.edu/???>" target="new"-GNU Emacs Navigator 등)를 통해 볼 수 있는 "<http://www.ucc.ie/xml/faq.html>" target="new"-HTML 버전도 제공됩니다.

XML 버전은 스펙이 동의되고, 그 문서를 다루는 브라우저들이 가능해지면 제공될 것이다.

일반문서(ASCII) 버전은 "<http://www.ucc.ie/doc/other/howtoftp.html>" target="new"-웹과 연결국은 여러 "<ftp://rtfm.mit.edu/>" target="new"-FTP 사이트에서 "<http://www.ucc.ie/doc/other/howtoftp.html>" target="new"-anonymous FTP를 통해 구할 수 있다. 이러한 버전은 "<http://www.ucc.ie/webmail/>" target="new"-WebMail 서버(전자우편을 통해 사용자들을 위한 서버)에 "<mailto:webmail@www.ucc.ie>"를 보내 얻을 수도 있다.

인쇄를 위해서는 PostScript™ 버전이 "<http://www.ucc.ie/xml/faq4.ps>" target="new"-A4와 "<http://www.ucc.ie/xml/faqlet.ps>" target="new"-Letter 사이즈로 제공됩니다.

인쇄물로 우송받기를 원하는 경우 "<mailto:silmaril@m-net.arbornet.com>"에 10불을 보내면 받을 수 있다. (정확한 액수와 주소를 위해 전자우편을 먼저 보내야 한다)

§ 일본어로 이 문서를 만든 Murata Makoto ("<http://www.fxis.co.jp/DMS/sgml/xml/xmlfaq.html>" target="new"-<http://www.fxis.co.jp/DMS/sgml/xml/xmlfaq.html>을 보라)와 스페인어로 번역해 준 Bilbao Deusto 대학의 Jaime Sagarduy ("<http://www.ucc.ie/xml/faq-es.html>" target="new"-<http://www.ucc.ie/xml/faq-es.html>을 보라)에게 감사드립니다.

"xml.gif"-XML logo와 아이콘이 "xml.ico"-ICO (Microsoft Windows), "xml_folder_icon.sit.hqx"-Mac, 그리고 "xml.xpm"-XBM (X Window system) 포맷으로 제공됩니다

질문들

"#FAQ-GENERAL"

"#FAQ-ACRO"-A.1 X

"#FAQ-DEF"-2 왜

"#FAQ-SGML"-A.3 SG

"#FAQ-HTML"-A.4 HT

"#FAQ-SAME"-A.5 XML, SGML, HT

"#FAQ-OWNS"-A.7 왜 XML은 SGML을 대체할 수 있는가

"#FAQ-IMPORT"-A.7 왜 XML은 SGML을 대체할 수 있는가

"#FAQ-IMPORT"-A.7 왜 XML은 SGML을 대체할 수 있는가

"#FAQ-EXTEND"-A.9 왜 XML은 SGML을 대체할 수 있는가

"#FAQ-WORD"-A.10 XML은 SGML을 대체할 수 있는가

"#FAQ-MORE"-A.11 XML은 SGML을 대체할 수 있는가

"#FAQ-MAILINGLIST"-A.12 XML은 SGML을 대체할 수 있는가

"#FAQ-USER"-B. 사용자 FAQ

"#FAQ-USEXML"-B.1 XML은 SGML을 대체할 수 있는가

"#FAQ-XMLOFFER"-B.2 왜 XML은 SGML을 대체할 수 있는가

"#FAQ-BROWSER"-B.3 XML은 SGML을 대체할 수 있는가

"#FAQ-SWITCH"-B.4 SGML을 XML로 대체할 수 있는가

"#FAQ-AUTHOR"-C. HTML은 XML을 대체할 수 있는가

"#FAQ-REPLACE"-C.1 XML은 SGML을 대체할 수 있는가

"#FAQ-XMLDOC"-C.2 XML은 SGML을 대체할 수 있는가

"#FAQ-SPACE"-C.3 XML은 SGML을 대체할 수 있는가

"#FAQ-CASE"-C.4 XML은 SGML을 대체할 수 있는가

"#FAQ-EXIST"-C.5 XML은 SGML을 대체할 수 있는가

"#FAQ-SUBSET"-C.6 XML은 SGML을 대체할 수 있는가

"#FAQ-LEARN"-C.7 HTML은 XML을 대체할 수 있는가

"#FAQ-CHARENTS"-C.8 XML은 SGML을 대체할 수 있는가

"#FAQ-DOCTYPE"-C.9 Document Type Definition은 XML을 대체할 수 있는가

"#FAQ-HYPertext"-C.10 XML은 SGML을 대체할 수 있는가

"#FAQ-MATH"-C.11 XML은 SGML을 대체할 수 있는가

"#FAQ-META"-C.12 XML은 SGML을 대체할 수 있는가

"#FAQ-JAVA"-C.13 XML, Active

"#FAQ-STYLE"-C

"#FAQ-DEVELOP

"#FAQ-SPEC"-

"#FAQ-VALIDWF"-D.2 `DTDless', `valid', `well-form

"#FAQ-WF"-D.2.1 `Well-formed' documents

"#FAQ-VALID"-D.2.2 Valid XML

"#FAQ-DTD"-D.3 SGML과

"#FAQ-XMLSOFT"-

"#FAQ-SWCHX"-D.5 X

"#FAQ-SSINCLUDES"-D.6 server-side INCLU

"#FAQ-CSINCLUDES"-D.7 client-side INCLU

"#FAQ-API"-D.9 X

답변들

A. 개론적 질문들

A.1 XML이란 무엇인가?

XML은 'Extensible Markup Language'의 약어이다(여기서 'extensible'이란 "#FAQ-HTML"-HTML과 같은 고정된 포맷이 아니라는 의미이다). XML은 웹에서 "#FAQ-SGML"-SGML을 사용하기 위해 고안된 것이다.

§ XML은 사실 약간은 부정확한 명칭이다 : XML 그 자체는 하나의 마크업 언어가 아니라 자신만의 독특한 마크업 언어를 고안할 수 있도록 해주는 메타언어(metalanguage)이기 때문이다. 보통 마크업 언어는 HTML의 예에서 보듯이 정보를 문서들의 특정한 클래스로 표현하는 방식을 정의한다. XML을 이용하면 자신만의 독특한 마크업 언어들을 정의할 수 있게 되므로 문서를 다양한 클래스들로 표현하는 것이 가능하다. 이는 XML이 SGML, 즉 마크업 언어들을 위한 국제 표준 메타언어로 선택되었기 때문이다.

A.2 왜 XML을 만들었는가?

XML은 웹상에서 "#FAQ-SGML"-SGML의 사용을 보다 쉽고 간단하게 하기 위해 고안되었다. 즉, 문서타입들의 정의를 쉽게하고, SGML로 정의된 문서들의 저작과 관리를 용이하게 하고, 그것들을 웹상에서 쉽게 전달하고 공유하기 위해 고안된 것이다.'

"#FAQ-SPEC"-XML 스펙에는 XML은 아주 쉽고 간단한 SGML의 방언(dialect)이며, XML의 목적은 포괄적인 SGML을 현재의 "#FAQ-HTML"-HTML처럼 웹상에서 서비스하고 수용하고 처리하는 것이다.'라고 정의되어 있다.

'이러한 이유로 XML은 SGML과 HTML 양자간의 상호운용성과 용이한 구현을 위해 고안되었다'("#FAQ-SPEC"-XML 스펙에서 인용).

A.3 SGML이란 무엇인가?

SGML이란 다양한 형식의 전자문서들의 구조와 내용을 묘사하는 방법을 정의한 국제적

표준인 "<http://www.sil.org/sgml/sgml.html>" target="new"-Standard Generalized Markup

Language ("<http://www.iso.ch/>" target="new"-ISO 8879)를 말한다. SGML에 관한

FAQ는 "<http://www.infosys.utas.edu.au/info/sgmlfaq.txt>" target="new"-

<http://www.infosys.utas.edu.au/info/sgmlfaq.txt>에 있고, SGML 웹페이지는 "<http://www.sil.org/sgml/>-

<http://www.sil.org/sgml/>에서 볼 수 있다.

A.4 HTML이란 무엇인가?

HTML은 "<http://www.w3.org/MarkUp>" target="new"-HyperText Markup

Language ("<http://ds.internic.net/rfc/rfc1866.txt>" target="new"-RFC 1866)의 약어이며, 웹 ("<http://www.w3.org/>"

target="new"-World Wide Web)에서 사용하는 "#FAQ-SGML" target="new"-SGML의 특정 어플리케이션 중 하나이다.

A.5 XML, SGML, HTML은 모두 다른 것인가?

물론 그렇다. "#FAQ-SGML"-SGML은 고대 수메리안의 문자표현에서부터 스텔스 폭격기의 기술문서에 이르기까지, 그리고 환자의 병원기록부터 음악 악보 표시까지, 인간행동의 많은 영역에서 사용되는 다양한 문서타입을 기술하는데 쓰이는 '母語(mother tongue)'이다.

"#FAQ-SGML"-HTML은 "<http://www.w3.org/>" target="new"-웹상에서 흔히 사용되는 이러한 문서 타입중의 하나에 불과하다. 그것은 제목, 문단, 목차, 그림 등과 하이퍼텍스트와 멀티미디어 지원을 통해 간단한 사무용 보고서들을 묘사할 수 있게 해주는 단 하나의 문서 타입이자 고정된 문서타입이다.

XML은 사용자만의 문서타입을 정의하고, 프로그래머가 프로그램을 쉽게 작성할 수 있도록 하기 위한 SGML의 축소된 버전이다. XML은 SGML의 복잡하고 덜 사용되는 부분을 생략함으로써, 어플리케이션의 작성을 쉽게하고 이해하게 쉽게 되어 있고, 웹상에서 전달과 처리(interoperability)에 더욱 적합하게 되어 있다. 그럼에도 XML은 여전히 SGML이며, XML 파일은 다른 어떤 SGML 파일과 마찬가지로 방법으로 파싱되며 유효화된다 ("#FAQ-XMISOFT"-XML 소프트웨어에 관한 질문들을 보라).

프로그래머의 입장에서는 XML을 HTML++, 즉 HTML의 확장으로 보기보다는 SGML로 보는 것이 유용할 것이다.

A.6 누가 XML에 대해 책임을 지고 있는가?

XML은 "<http://www.w3.org/>" target="new"-W3C(World Wide Web Consortium) 프로젝트 중 하나이다. 그리고 W3C의 XML 워킹그룹(Working Group)이 스펙의 개발을 관리한다. 다양한 분야에서 선출되어 구성된 SIG(Special Interest Group)는 전자우편을 통해 의견을 내고 검토를 한다.

XML은 하나의 공식 포맷이다. 이것은 어느 하나의 회사에서 소유하는 개발이 아니다.

A.7 왜 XML이 그렇게 중요한 개발인가?

XML은 웹 개발자들을 가로막는 두 가지 속박을 없앤다.

1. 단일하고 고정적인 문서타입("#FAQ-HTML"-HTML)에의 의존
2. 강력하지만 까다로운 프로그램 옵션을 허용하는 구문을 가진 "#FAQ-SGML"-SGML의 복잡성.

XML은 SGML에서의 선택사항(optionality)의 단계를 간소화하고, 웹에서 사용자정의 문서형식의 개발을 가능하게 한다.

A.8 어떻게 XML은 SGML을 간략히 하고, 그럼에도 어떻게 독자적 문서형식을 정의할 수 있도록 하는가?

SGML을 간략히 하기 위해 XML은 "#FAQ-SGML"-SGML의 내부 값(internal values) 파라미터(parameters)들 중 일부를 재정의 하고, 프로그램하기 어렵게 만드는 아주 많은 복잡하고 거의 사용되지 않는 특징들을 삭제하였다("#FAQ-SPEC"-XML 스펙의 부록 A를 보라).

그러나 XML은 당신의 고유한 문서타입을 정의할 수 있도록 하는 SGML의 구조적 특징 모두를 유지한다. 그것은 또한 미리 정의된 문서타입을 요구하지 않는 새로운 형태를 도입하였다. "#FAQ-VALIDWF"-`valid`와 `well-formed` documents에 관한 질문들과 "#FAQ-DEVELOPER"-개발자를 위한 질문들에 있는 "#FAQ-SELFDEF"-어떻게 당신의 고유한 문서형식을 정의할 것인가를 보라.

A.9 왜 HTML의 확장만으로 해결할 수 없는가?

"#FAQ-HTML"-HTML은 정보를 묘사하는데 단 하나의 방법만을 제공하기 때문에 각 회사에서 내놓고 있는 재미있지만 종종 호환되지 않는 확장들로 인해 이미 한계에 도달하였다.

XML은 사람들과 조직들이 각 영역(음악, 화학, 전기공학, 등산, 재정, 서핑, 어학, 수학, 뜨개질, 역사, 공학, 토끼 기르기 등)에서 정보를 교환하기 위한 독자적 마크업 언어를 만드는 것을 가능하게 할 것이다.

HTML은 정보를 기술하는 방법으로서 유용함은 한계에 달하였다. 그리고 비록 그것이 앞으로도 현재와 같이 콘텐츠를 위한 중요한 역할을 계속한다고 할지라도 많은 새로운 어플리케이션들은 더 강력하고 유연한 기반(infrastructure)을 요구하고 있다.

A.10 왜 우리는 SGML의 모든 것을 필요로 하는가? WORD 나 NOTES 를 사용하면 안되는가?

서로 다른 종류의 컴퓨터들이 연결되어 있는 네트워크상의 정보는 모든 컴퓨터에서 이용가능해야 한다. 공공 정보는 특정 메이커나 모델, 업체에 제한될 수 없으며 데이터 포맷의 통제권을 개인에게 넘길수 없다. 그러한 정보는 또한 여러가지 방법으로 이용될 수 있는 하나의 품으로 만드는 것이 유용하다. 이러한 방법은 시간과 노력의 낭비를 줄일 수 있을 것이다.

"#FAQ-SGML"-SGML은 이런 종류의 어플리케이션을 정의하는데 사용되는 국제 표준이다. 그러나 서로다른 소프트웨어에 기반한 대안을 필요로 하는 사람들은, 특히 사적인 이용을 위해서라면 SGML을 이용하여 유사한 서비스를 개발하는데 전적으로 자유롭다.

A.11 XML에 관한 상세정보는 어디에서 구할 수 있는가?

온라인에는 W3C에서 제공한 "#FAQ-SPEC"-XML 스펙과 보조적인 문서들이 있다. 여기에는 Robin Cover가 쓴 "<http://www.sil.org/sgml/sgml.html>" target="new"-SGML pages안에 들어있는 "<http://www.sil.org/sgml/xml.html>" target="new"-XML section과 온라인 참조물 리스트, Tim Bray가 작성한 "<http://www.textuality.com/xml/>"

[target="new"-summary](#)와 <http://www.textuality.com/xml/faq.html> [target="new"-condensed FAQ](#)가 있다.

아래의 목록은 관리자가 발견할 수 있었던 것들이다. 만약 당신이 다른 것들을 발견한다면 나에게 <mailto:pflynn@imbolc.ucc.ie>-**메일**을 주기 바란다. 오래된 것들은 참조하기 위해 잠시 여기에 보류되어 지겠지만 결국은 삭제될 것이다.

- § Technology Appraisals사는 영국 런던에서 1998년 4월 6일부터 8일까지 *XML ready for prime time?*이라는 주제로 세미나를 개최한다. 상세정보는 TAL의 <mailto:techapp@cix.compulink.co.uk>-**Susan Dennington**에 있다.
- ¶ James G. Miller & Associates는 1998년 4월 14일 <http://www7.conf.au/> [target="new"-www7](#)에서 하루종일 XML에 관한 워크숍을 진행할 것이다.
- ¶ www.vsms.nottingham.ac.uk/vsms/java [target="new"-www7](#)에서 *Scientific Information Components using Java and XML*라는 표제로 <http://www.vsms.nottingham.ac.uk/vsms/java/advert/advert.txt> [target="new"-XML/Java Virtual Course](#)를 준비하고 있다. 상세정보는 <http://www.vsms.nottingham.ac.uk/vsms/java/advert/advert.txt> [target="new"-http://www.vsms.nottingham.ac.uk/vsms/java/advert/advert.txt](#)에 있다. 여기서 XML은 DTD나 엔터티(entity), 선언부분(marked section), 카탈로그, 링크 등이 없는 *예를 들어* 단순한 **#FAQ-WF**-**well-formed**, 태그하는 법, 속성 부여하는 법 등 가장 기초적인 것들이 다루어질 것이다. 그것은 (기존 문서로부터 추출해내는 법을 포함하여)엘레먼트 트리(element trees)를 만드는 방법을 중심으로 다를 것이다.
- ± 그래픽 커뮤니케이션 협회 (Graphic Communications Association; GCA)에 의해 매년 열리는 SGML 컨퍼런스는 SGML/XML 컨퍼런스로 이름이 바뀌었다. SGML/XML '97은 1997년 12월 8일부터 11일까지 Washington DC에서 열렸었다 (상세정보는 <http://www.gca.org/> [target="new"-GCA의 웹사이트에서 찾을 수 있다](#)).

§ Robin Cover의 <http://www.sil.org/sgml/sgml.html> [target="new"-SGML pages](#)에는 XML에 관한 책과 기사 목록이 있다.

A.12 XML의 응용과 개발에 관한 토론은 어디에서 할 수 있는가?

XML 구성요소(components) 개발을 위해 구성된 xml-dev라는 메일링리스트가 있다. <mailto:majordomo@ic.ac.uk>-**majordomo@ic.ac.uk**에 다음과 같은 한 줄짜리 메일을 보내면 가입할 수 있다.

```
subscribe xml-dev yourname@yoursite
```

이 메일링리스트는 <http://www.lists.ic.ac.uk/hypermail/xml-dev/> [target="new"-http://www.lists.ic.ac.uk/hypermail/xml-dev/](#)에서 하이퍼메일형태로 제공되어 웹상에서 볼 수도 있다.

이 메일링리스트는 XML에 관한 개발 리소스에 직접적으로 연관된 사람들을 위한 것이라는 것을 염두에 두길 바란다. 이것은 XML에 관한 일반정보를 위한 것이 *아닐 뿐더러* (이 FAQ를 보거나 **#FAQ-MORE**-**다른 자료들**을 보길 바란다) SGML 응용과 자료에 관한 일반적인 토론 (이를 위해서는 news:comp.text.sgml-**comp.text.sgml**을 보라)을 위한 것도 아니다.

공개토론을 위해 범용적으로 이용할 수 있는 XML-L이라는 메일링리스트가 있다. 여기에 가입하려면 <mailto:listserv@listserv.he.a.ie>-**LISTSERV@listserv.he.a.ie**에

```
subscribe XML-L forename(이름) surname(성)
```

과 같은 형식으로 한 줄짜리 메일을 보내면 된다 (당신의 실제 이름과 성으로 바꾸어 보내야 한다).

탈퇴하기 위해서는 같은 주소로

unsubscribe XML-L

라고 써서 보내면 된다.

당신이 어떠한 메일링리스트에 가입하였을 때 받게 되는 설명문(Fine Documentation)을 읽기 바란다. 그 문서는 특히 당신의 전자우편 주소가 바뀌었을 때 어떻게 해야하는 지 등 중요한 정보를 담고 있다.

B. (HTML 브라우저 이용자를 포함하여) SGML 이용자를 위한 질문들

B.1 XML을 사용하기 위하여 무언가를 해야만 하는가?

아직은 그렇지 않다. XML은 아직 개발중에 있으나, 이미 "#FAQ-BROWSER"-몇몇 선구적인 브라우저들이 존재하므로 당신은 그것들로 실험해 볼 수 있다. 스펙이 완전해지면 더 많은 프로그램들이 출현하기 시작할 것이며, 당신은 현행의 어플리케이션을 사용하듯이 브라우저를 다운받아 웹을 브라우징하기 위해 그것들을 사용할 수 있을 것이다.

"<ftp://sunsite.unc.edu/pub/sun-info/standards/xml/eg/>" target="new"-존 보삭(Jon Bosak)의 셰익스피어 연극대본(Shakespeare plays)과 "<http://www.venus.co.uk/omf/cml/>" target="new"-Chemical Markup Language (CML)을 이용하는 분자실험 등 막 출현하는 XML 자료들을 보기 위해서 당신은 선구적 브라우저들을 사용할 수 있다."<http://www.sil.org/sgml/xml.html#examples>" target="new"-<http://www.sil.org/sgml/xml.html#examples>에 더 많은 사례들이 열거되어 있다.

당신 자신의 XML 문서를 작성하기 위한 준비를 시작하고 싶다면 "#FAQ-AUTHOR"-저자를 위한 질문들을 참조하라.

B.2 왜 HTML 대신 XML을 사용해야 하는가?

- 저자와 문서 공급자들은 HTML로 허덕이는 대신에 XML을 이용하여 "#FAQ-DOCTYPE"-그들 자신의 문서타입을 설계할 수 있다. 문서 타입은 정확하게 읽는 사람들에게 맞추어 질 수 있다. 그래서 "#FAQ-HTML"-HTML의 경우 특정한 효과를 내기위해 필연적으로 발생하는 성가신 작업들은 이제 과거의 유물이 되어야 한다. 저자나 디자이너는 자유롭게 그들 자신의 마크업 요소(element)들을 고안하게 될 것이다.
- 정보의 내용은 보다 풍부하고 사용하기 쉬워질 것이다. 왜냐하면 "#FAQ-HYPertext"-XML의 하이퍼링크 기능이 HTML보다 훨씬 크기 때문이다.
- XML은 브라우저의 표현과 수행을 위해 더 풍부하고 나은 기능들(facilities)을 제공할 수 있다.
- XML은 보다 유연한 모델을 지원하여 SGML의 기저에 있는 복잡성의 많은 부분들을 제거한다. XML을 다루는 프로그램을 작성하는 것은 완전한 SGML 프로그램을 작성하는 것보다 훨씬 용이해질 것이다.
- 정보에 대한 접근과 재사용이 보다 용이해질 것이다. 왜냐하면 HTML의 경우에서와 같이 특정 회사에 의해 제한되는 대신에, 어떠한 XML 소프트웨어를 사용하여도 XML의 보다 유연한 마크업이 사용될 수 있기 때문이다.
- "#FAQ-VALID"-유효한 XML 파일들(valid XML files)은 적법한 SGML이다. 따라서 그것들은 스펙이 안정화되고, SGML 소프트웨어가 그것을 채택하기만 하면, 웹 이외의 SGML환경에서도 사용될 수 있다.

B.3 XML 브라우저는 어디에서 구할 수 있는가?

§ 이미 몇몇 브라우저들이 출현하였지만 "#FAQ-SPEC"-XML 스펙은 여전히 새로운 것이다. "#FAQ-HTML"-HTML과 같이 단지 하나의 브라우저가 아니라 여러 개의 브라우저가 등장할 것이다. 어쨌든 수많은 서로 다른 XML 어플리케이션이 등장할 수 있다는 잠재력은 어떠한 브라우저도 모든 것을 다 처리할 수 없다는 것을 의미한다.

예를 들어 파싱, 트리 관리, 검색, 포매팅 등 XML의 일반적 부분은 "#FAQ-DEVELOPER"-개발자들이 XML 어플리케이션을 작성할 때 일관성있게 작업할 수 있도록 범용의 브라우저 라이브러리나 툴킷에 통합되고 있다. 이후에 이러한 어플리케이션들은 의미(semantics)를 추가함으로써 특정 분야에 맞추어 질 수 있거나 자바와 같은 랭귀지를 사용하여 일반 브라우저를 위해 플러그인 프로그램을 개발할 수 있을 것이다. 또한 웹상에서의 정확한(transparently) 전송을 위한 특별한 모듈도 가질 것이다.

§ Netscape와 Microsoft는 지금 둘다 XML 기능(facilities)을 개발하고 있다. Microsoft의 개발현황은 "<http://www.microsoft.com/msdn/sdk/inetsdk/help/> target="new"-

<http://www.microsoft.com/msdn/sdk/inetsdk/help/>에서 볼 수 있다. Netscape의 진행상황은 현재 알려져 있지 않다.

§ "#FAQ-XMISOFT"-저자와 "#FAQ-API"-개발자를 위한 소프트웨어 기록을

참조하라. "<http://www.sil.org/sgml/xml.html> target="new"-<http://www.sil.org/sgml/xml.html>에 있는 SGML 웹사이트 내에 XML 파트를 보면 더 많은 리스트를 구할 수 있다.

B.4 SGML이나 HTML을 XML로 변환해야 하는가?

그렇지 않다. 현재의 "#FAQ-SGML"-SGML과 "#FAQ-HTML"-HTML 어플리케이션 소프트웨어는 계속하여 현존하는 파일들을 다룰 것이다. 그러나 확장된 기능들을 가지고 XML 파일을 보거나 다운받기를 원한다면, 그리고 그것을 이용하기 원한다면 XML을 지원하는 소프트웨어가 나올 때 그것을 추가할 필요가 있을 것이다.

C. HTML을 쓰는 사람을 포함하여 SGML 저작자들을 위한 질문

저자들은 XML 파일의 내부에 관한 더 많은 정보를 담고 있는 "#FAQ-DEVELOPER"를 읽어야 한다.

C.1 XML은 HTML을 대체하는가?

그렇지 않다. XML은 그 자체로 "#FAQ-HTML"-HTML을 대체하지 않는다. 대신에 사용자가 자신만의 마크업 요소(element)를 정의할 수 있도록 함으로써 대안을 제공한다. HTML은 앞으로도 얼마간 일반적으로 사용될 것이며, DTD들은 원래의 SGML 버전 뿐 아니라 XML 버전으로도 사용될 것이다. XML은 완벽한 "#FAQ-SGML"-SGML보다 더욱 단순하게 DTD를 작성할 수 있도록 설계되었다.

HTML의 XML 버전과 대중적인 DTD들을 만드는 작업이 계속되고 있다. 그러나 이러한 작업은 1997년 11월을 목표로 하는 XML 1.0 스펙이 끝날 때까지 완료되지 못할지도 모른다. 이에 관한 공지는 comp.text.sgml과 XML-L를 주목하라.

C.2 XML 문서의 내부는 어떻게 생겼는가?

기본적인 구조는 HTML을 포함하여 다른 대부분의 SGML 어플리케이션과 유사하다. XML 문서는 문서 타입 선언(document type declaration)없이 저자 자신이 설계한 마크업안에 문서를 집어넣어 아주 간단히 작성될 수도 있다.

```

"#FAQ-RMDPI"-<?XML version="1.0" standalone="yes"?>
<conversation>
  <greeting>Hello, world!</greeting>
  <response>Stop the planet, I want to get off!</response>
</conversation>

```

또는 특정한 DTD와 내부의 부분정의(internal subset), 그리고 보다 복잡한 구조를 가지는 복잡한 문서일 수도 있다.

```

<?XML version="1.0" "#FAQ-RMD"-standalone="no" encoding="UTF-8"?>
<!"#FAQ-DOCTYPE"-DOCTYPE titlepage SYSTEM "http://www.frisket.org/dtds/typo.dtd"
[<!ENTITY % active.links "INCLUDE">]>
<titlepage>
  <white-space type="vertical" amount="36"/>
  <title font="Baskerville" size="24/30"
    alignment="centered">Hello, world!</title>
  <white-space type="vertical" amount="12"/>
  <!-- In some copies the following decoration is
    hand-colored, presumably by the author -->
  <image location="http://www.foo.bar/fleuron.eps" type="URL" alignment="centered"/>
  <white-space type="vertical" amount="24"/>
  <author font="Baskerville" size="18/22" style="italic">Munde Salutem</author>
</titlepage>

```

또는 XML 문서들은 두 가지 사이에서 어떻게든 사용될 수 있다. 많은 것들이 스스로의 문서 타입이 어떻게 정의되기를 원하는지(또는 누구의 것을 사용하기를 원하는지)와 문서타입이 무엇을 위해 사용되는지에 달려있다. "#FAQ-VALIDWF"-**유효한 파일과 잘 정리된 파일(valid and well-formed files)**에 관한 질문들을 참조하라.

C.3 XML은 문서 내의 공백을 어떻게 표현하는가?

공백과 관련한 SGML 규칙들은 XML을 위해 변경되어왔다. 그래서 줄바꿈(linebreak), 탭(TAB) 문자, 정식 띄어쓰기(regular space) 등 모든 공백은 파서에 의해서는 **변경되지 않은 채** 브라우저, 포맷터, 뷰어 등의 어플리케이션에 넘겨진다. 이것은 다음을 의미한다.

- § 구조적 요소들(element) 사이의 '무의미한' 공백은 (종종 '요소 내용(element content)'라 불리우는 텍스트 데이터가 아니라 단지 다른 요소들만 포함하고 있는 요소들) 어플리케이션에 넘겨질 것이다 (완전한 SGML의 경우 이러한 공백은 삭제된다).
- § 텍스트와 마크업이 함께 섞여 있는 형태('mixed content 또는 PCDATA[parsed character data])를 포함할 수 있는 요소(element) 내의 '의미있는' 공백은 이전과 마찬가지로 어플리케이션에 넘겨질 것이다.

```

<chapter>
  <section>
    <title>
      My title for Section
    1.
  </title>
  <para>
    ...
  </para>
</section>
</chapter>

```

파서는 어쨌든 요소 내용 내에 어떤 종류의 공백이 있는지 어플리케이션에 넘겨주어야 한다 ('완벽한' SGML 이용자들은 이러한 정보가 "<http://www.sil.org/sgml/WG8-n931a.html>" target="new"-ESIS내에는

없고, "<http://www.sil.org/sgml/topics.html#groves>" target="new"-grove 내에, *있다*는 것을 알고 있을 수도 있다.). 위의 예를 보면 어플리케이션은 section title 내에 보이는 것 뿐 아니라 요소들 사이의 잘 표현된 줄바꿈, 탭, 띄어쓰기를 모두 넘겨받을 것이다. 어떤 종류의 공백을 버리고 또는 유지할지 결정하는 것은 브라우저, 포맷터, 뷰어 등 어플리케이션의 기능이다.

C.4 XML 문서 중 어떤 부분이 대소문자 구별을 하는가?

§ XML 파일은 전체가 다 대소문자 구별을 한다. 이것은 마크업 *뿐 아니라* 텍스트에도 해당된다. 이것은 HTML을 포함한 다른 많은 SGML 문서형식과의 중대한 차이이다. 비영어권 문자 스크립트를 이용한 마크업을 허용하는 것과 대소문자 구별을 하지 않는 스크립트 내의 문제점들을 제거하는 것이 도입되었다.

- 시작 태그와 마침 태그에 사용되는 요소의 이름들(element names)은 대소문자 구별을 한다. 요소의 이름을 정의하기 위해 섞어서 사용하는 대소문자를 끝까지 유지해야 한다. 스스로 정의하는 경우나 "#FAQ-DTD"-DTD를 사용하는 경우 모두 해당된다.
- DTD가 없는 well-formed 파일들은 요소의 이름이 *처음 나타날 때* 대소문자 구별을 정의한다. 따라서 <BODY>... </body>와 같은 형식으로 사용할 수 없다. 대소문자는 반드시 일치하여야 한다. 따라서 와 는 *두 개의 서로 다른 요소*이다.
- 요소에 기반을 둔 속성의 이름(attribute names)도 마찬가지이다. 예를 들어 동일한 파일 내에 있는 <PIC width="7in"/>와 <PIC WIDTH="6in"/>는 *두 개의 별개의 속성*이다. 왜냐하면 소문자로 작성된 width와 대문자로 작성된 WIDTH는 서로 구별된다.
- 속성값(attribute values)들도 또한 대소문자 구별을 한다. 문자 데이터 값은 *예를 들면* HRef="MyFile.SGML"와 같이 이전과 정확히 같다. 그러나 ID와 IDREF 속성(attribute)은 대소문자 구별을 하며 비교를 위해 더이상 대문자를 고집하지 않아도 된다.
- 모든 엔티티(entity)의 이름들(Á)과 데이터의 내용은 대소문자 구분을 하며, 이런 점은 예전과 마찬가지이다.

C.5 어떻게 HTML 파일을 XML에서 작동하게 할 수 있는가?

§ HTML 문서들을 "#FAQ-WF"-**잘 정리되게 즉, well-formed**하게 만들면 된다. 아래를 보라. XML에서 DTD는 선택사항이다. 그러나 HTML 파일들은 HTML DTD의 XML 버전이 아직 없기 때문에 ("**#BEN**"-**진행 중임**) 현재 어디에서도 DTD가 없을 수 밖에 없다. 먼저 존재하는 HTML 파일들을 well-formed하게 만드는 것이 필요하다. 왜냐하면 대부분의 HTML DTD들은 마침 태그의 생략을 허용하지만 XML의 경우는 그러하지 않기 때문이다. 이미 많은 HTML 저작도구들은 아주 충분하지는 않지만 거의 well-formed한 XML을 만들어낸다.

모든 XML 문서들은 well-formed하여야 하지만(아래를 참조하라), DTD는 조건부이다. HTML 파일들은 XML의 DTD가 없는 형식으로 변환될 수 있다. 그러나 현재 SGML의 HTML DTD들의 XML 버전은 있을 수 없다. 그리고 많은 HTML 저작도구들은 충분하지는 않지만 거의 well-formed한 DTD없는 XML을 만들어 낸다.

만약 당신이 여러 개의 HTML "**#FAQ-DTD**"-Document Type Definitions (DTDs) 중 하나를 따르는 "**#FAQ-HTML**"-HTML 파일을 만들었고 그것들이 유효하다면(validate OK), 그것들은 다음과 같이 XML로 변환될 수 있다.

- § DOCTYPE 선언(declaration)과 내부의 부속항목들(internal subset)--이는 기본적으로 <!DOCTYPE HTML...>의 껍질 안에 들어 있는 모든 것들이 해당된다--을 <?XML version="1.0" standalone="yes"?>과 같은 XML 선언으로 바꾼다.
- 모든 독립요소(EMPTY elements)들의 오른쪽 껍질을 `>`--예를 들면 로 바꾼다. 독립요소들로는 다음과 같은 것들이 있다. 문서의 헤더(header)에서 사용되는 <ISINDEX>, <BASE>, <META>, <LINK>, <NEXTID>, <RANGE>와 문서의 바디(body)에서 사용되는 ,
, <HR>, <FRAME>,

<WBR>, <BASEFONT>, <SPACER>, <AUDIOSCOPE>, <AREA>, <PARAM>, <KEYGEN>, <COL>, <LIMITTEXT>, <SPOT>, <TAB>, <OVER>, <RIGHT>, <LEFT>, <CHOOSE>, <ATOP>, <OF>가 있다.

- 마침 태그가 필요한 요소들(non-empty elements)이 정확히 사용되었는지 확인하라. 예를 들어 <P>는 항상 </P>로 닫혀져야 한다. 이 작업은 "<http://www.jclark.com/sp/>" target="new"-SP의 일 부분인 *sgmlnorm*과 같은 표준화된 프로그램이나 *Emacs/psgm*과 같은 에디터의 *sgml-normalize* 기능에 의해 자동적으로 처리될 수 있다.
- 마크업에 사용되는 문자인 <과 &를 <와 &로 바꾼다.
- 모든 속성값(attribute values)이 따옴표 안에 있는지 확인한다.
- 요소의 이름들이 나올 때 시작태그 & 마침태그에서 대소문자 구별이 정확히 일치하는지, 그리고 전체문서에서 일관적으로 사용되는지 확인한다.
- 모든 속성의 이름들이 전체문서를 통해 일관적으로 사용되는지 확인한다.

HTML 브라우저들은 XML 스타일의 독립요소(EMPTY elements)들의 종료처리(trailing slash)를 이해하지 못하기 때문에 위와 같이 변환된 것은 반대로는 호환되지 않는다. 하나의 대안은 를 로 바꾸는 것과 같이 모든 독립요소에 가짜 마침 태그를 붙이는 것이다.

만약 많은 양의 유효한(valid) HTML 파일들을 가지고 있다면, 이러한 작업을 SGML 변환 시스템-- "<http://www.omnimark.com/>"-*Omnimark*, "<http://www.balise.com/>"-*Balise*, "<http://www.dircon.co.uk/sgml/>"-*SGMLC*, 또는 *SGML Perl* 라이브러리 사용하는 시스템 등이 있다--에서 스크립트를 이용하여 처리할 수도 있으며, 하고자 하는 작업을 잘 알고 있다면 에디터의 매크로 기능을 이용해서도 처리할 수 있다.

만약 HTML 파일들이 유효하지 않다면(invalid) 그것들은 거의 수작업으로 처리해야 하는 것이 확실하다. 잘못된 부분이 규칙적이고 조심스럽게 작성되었다면 그 파일은 실질적으로 거의 *well-formed*할 것이며, 위와 같은 작업을 하는 프로그램이나 스크립트를 작성할 수 있지만 유효한지 여부와 적합여부를 판단하기 위해서는 다음을 확인해 보라.

- § 파일에 마크업 문법상의 오류는 없는가? 예를 들면 슬래쉬(/) 대신 역슬래시가 사용된 것은 없는지, 또 요소들이 부정확하게 중첩되어 사용된 곳은 없는가? 예: <SAMP>요소가 안에서 시작되었는데 </SAMP> 이 밖에서 선언되지 않았는가?
- 목록 항목(list item) 안에 제목을 사용하거나 목록 환경(list environments) 밖에서 목록 항목을 사용하는 것과 같이 HTML DTD와 모순되게 마크업을 사용하지 않았는가?
- § 파일에 어떤 DTD에도 없는 요소가 사용되지 않았는가? 비록 이것이 DTD없는 *well-formed* 파일로 변환하는데 쉽지만--미리 요소를 정의하지 않아도 되기 때문에--브라우저에 따라 특별한 확장들은 결코 공식적으로 정의된 적이 없기 때문에 이러한 방법이 어디에서나 의미있게 사용될 수 있는지 확인하는 것은 불가능하다.

유효하지만 의미가 없거나 쓸모없는 마크업은 변환 전에 수정되어야 할 지도 모른다. 예를 들어 반복되는 빈 문단들, 줄바꿈, 비어있는 테이블, 공백을 넣기 위해 사용된 그림(invisible 'spacing' GIFs) 등은 XML에서는 스타일시트를 사용하기 때문에 필요가 없다.

XML에서 변환시 필요한 것들의 자세한 사항은 "[#FAQ-WF](#)"-'*well-formed*' XML 파일의 규칙들을 참조하라.

HTML DTD의 XML 버전이 준비 중임을 유념하라.

- "<mailto:btrafford@worldnet.att.net>"-*Ben Trafford*가 HTML 4.2의 XML 버전을 개발하고 있다.
- [찾을 수 있는 다른 항목들에 대해서는 "<mailto:pflynn@m-net.arbornet.org>"-*편집자*에게 문의하라.]

C.6 XML이 SGML의 부분집합이라면, SGML 툴을 이용하여 직접 XML을 다룰 수 있는가?

다음의 조건이 충족된다면 가능하다. a) 문서가 유효한 "#FAQ-DTD"-Document Type Definition (DTD)를 가지고 있어야 한다. 다시말해 파일들은 단지 "#FAQ-WF"-well-formed할 뿐 아니라 "#FAQ-VALID"-valid하여야 한다. b) XML을 지원하기 위해 필요한 특징들--독립요소(EMPTY element)를 위한 특별한 형식, NAMECASE GENERAL NO와 같은 SGML 선언(declaration) 중 일부, 복수 속성 선언(multiple attribute declarations)등--을 알고 있는 소프트웨어를 사용해야 한다.

지금 당장은 이러한 독립요소(EMPTY elements)의 형식 때문에 XML 파일을 다루는 툴들이 거의 없지만, 변화하고 있는 것은 사실이다. nsgmls파서는 실험적인 XML 변환기능을 가지고 있으며, 최초의 XML 전용 에디터와 파서들이 등장하고 있다("#FAQ-XMISOFT"-소프트웨어에 관한 질문들을 참조하라).

ISO 8879의 규약들의 부분 개정 논의가 되고 있으며, 이들 중 일부는 웹을 지원하는 데 필요한 기능 변화에 관한 것들이다.

C.7 HTML에 익숙하면, 쉽게 XML을 배울 수 있는가?

당연히, 매우 쉬운 일이다. 그러나 지금 현재에는 여전히 튜토리얼, 간단한 도구, 더 많은 XML 문서의 예가 필요하다. "#FAQ-WF"-Well-formed XML 문서는 매우 중요하지만 아주 적은 부분의 구문을 제외하고는 "#FAQ-HTML"-HTML과 유사하게 보일 것이다.

모든 사용자 집단은 그들만의 독자적으로 정의된 문서형식을 갖을 수 있기 때문에 XML을 배우는 것은 무척 쉬운 것이다. 이것은 요소이름을 적절히 골라서 붙일 수 있기에 가능하다.

C.8 XML은 비영어권 문자를 사용할 수 있는가?

그렇다. "#FAQ-SPEC"-XML 스펙을 보면, XML은 모두 인류의 문자를 포함하는 국제표준인 31-bit 문자집합인 "<http://www.iso.ch/>" target="new"-ISO 10646을 사용한다고 나와있다. 이것은 현재 유니코드(Unicode)와 일치한다.

§ 스펙 2장 2절을 보면, '모든 XML 프로세서는 ISO 10646의 UTF-8과 UTF-16 인코딩을 받아들여야 한다...'. UTF-8은 유니코드의 8비트 문자로의 인코딩이다. 처음의 128개는 아스키(ASCII)와 동일하며, 나머지는 유니코드의 나머지 부분을 2바이트에서 6바이트까지 차례로 인코드하기 위해 사용된다. 단일한 8중주 형태인 UTF-8은 따라서 ISO 646 IRV(ASCII)과 동일하다. 그러므로 영어를 위해 아스키를 사용하거나 영어권 알파벳을 이용하여 악센트없는 문자를 사용할 수 있다. UTF-8은 아스키의 마지막인 10진으로 126번 코드 이후로는 ISO 8859-1(ISO Latin-1)과는 호환되지 않는다는 점을 유의하라. UTF-16은 UTF-8과 비슷하지만, 64,000 문자의 다음 16단계를 두 개의 16비트 문자들로 대표하려는 계획을 가지고 있다.

'... 둘 중에 어떤 것이 사용되고 있지 알려주는 메커니즘과 다른 인코딩이 사용될 수 있도록 하는 메커니즘들이[...]. 문자인식에 관한 토론 중에 이야기되고 있다.'"#FAQ-SPEC"-XML 스펙은 XML 파일에서 어떤 코드의 문자세트를 당신이 사용하고 있는가를 상술하는 법을 설명하고 있다.

§ USC-4의 사용은 단지 ISO 8879에 계류중인 'WebSGML Adaptations'이 8자리가 넘는 숫자들이 SGML 선언(Declaration)에서 사용될 수 있도록 효력을 발생하게 되었을 때만 SGML과 XML에서 적법하게 상술될 수 있다.

§ '사용되는 특별한 인코딩에도 불구하고, ISO 10646 문자세트 내의 어떠한 문자도 10진 혹은 16진의 대응하는 숫자 코드를 가리킬 수 있다.' 따라서 어떠한 문자세트를 개인적으로 사용한다고 하더라도 (10진문자 코드의 경우) &#ddd;를 사용하거나 (16진 코드의 경우 대문자로) &#UHHHH;를 사용함으로써 인코드 문자저장소의 특정한 개개의 문자들을 여전히 가리킬 수 있다.

용어와 숫자들이 혼란스럽게 할 수도 있을 것이다. "

[10646.html" target="new"-ISO 10646 개념사전\(ISO 10646 Concept Dictionary\)](#)을 참조하라.

C.9 DTD란 무엇이고, 어디에서 구할 수 있는가?

DTD는 문서의 특별한 형식의 공식적 정의를 담고 있는 하나의 파일(또는 함께 사용되는 복수의 파일들)이다. DTD는 요소들의 이름, 요소들이 언제 나타나는지, 어떻게 함께 사용되는지 등을 정한다. 예를 들어 <ITEM>를 포함하는 <LIST>를 기술하는 문서형식을 원한다면 DTD의 일부는 다음과 같은 것을 포함할 것이다.

```
<!ELEMENT item (#pcdata)>
<!ELEMENT list (item)+>
```

DTD는 텍스트를 포함하는 아이템과 아이템들을 포함하는 리스트를 정의한다. DTD는 프로세서가 자동적으로 문서를 파싱하고, 모든 요소들이 언제 나오고 어떻게 서로 연관되어 있는지 구분할 수 있도록 하는 공식적인 언어이다. 그러므로 스타일시트, 내비게이터, 브라우저, 검색엔진, 데이터베이스, 인쇄루틴 및 다른 어플리케이션들을 사용할 수 있도록 한다.

[XML의 경우 요소 이름과 콘텐츠 모델 사이에서의 요소 정의에는 '-'나 'O' 문자(즉, minimization parameter)가 사용되지 않는다는 점을 유의하라. 왜냐하면 독립요소를 제외한 모든 요소들은 모든 경우에 시작 태그와 마침태그가 나와야 한다.]

모든 분야에 수천개의 뛰어난 SGML DTD가 이미 존재한다. (예를 들면 "[http://www.sil.org/sgml/sgml.html" target="new"-SGML Web pages](http://www.sil.org/sgml/sgml.html)를 보라.) 그것들 중 많은 것들을 다운받아 공짜로 사용할 수 있다. 또한 스스로 만들 수도 있다. 다른 언어들과 마찬가지로 DTD를 만들기 위해서는 먼저 배워야 한다. 무엇이 사용되면 안되는가 하는 "#FAQ-RESTRICT"-**제한 목록**을 참조하라. 하지만 XML은 SGML보다는 몹시 간단하다. 현존하는 SGML DTD가 XML 시스템에서 사용되기 위해서는 XML로 변환되어야 한다. 유명한 DTD들이 XML 형식으로 제공된다는 소식을 기대하라.

C.10 XML에서 링크는 어떻게 처리하는가?

"#TEI-LINK"-XML 시스템의 링크기능은 HTML보다 훨씬 강력하다. 현재의 HREF 스타일의 링크는 여전히 사용가능할 것이다. 새로운 링크 기술은 하이퍼텍스트를 포함하는 다른 표준의 개발들--예를 들면 "[http://www-tei.uic.edu/orgs/tei/" target="new"-TEI](http://www-tei.uic.edu/orgs/tei/)와 "[http://www.sil.org/sgml/" target="new"-HyTime](http://www.sil.org/sgml/), 이러한 기술들은 특정 문서보다는 당신의 문서나 다른 사람의 문서의 일부분을 링크할 수 있도록 하고, 또한 양방향 링크나 다방향 링크도 제공한다--로부터 배운 성과들에 기반하고 있다. 이러한 것들은 *Panorama*와 *Multidoc Pro* 같은 SGML 브라우저에는 이미 구현되어 있다.

"[http://www.w3.org/TR/WD-xml-link" target="new"-XML Linking Specification \(XLL\)](http://www.w3.org/TR/WD-xml-link)문서는 구체적인 스펙을 보여준다. XML의 링크는 URL일 수도 TEI 스타일의 Extended Pointer("#TEI-LINK"-Xptr)일 수도 있다. 혹은 둘다도 가능하다. URL은 그 자체로 하나의 자원으로 여겨진다. Xptr이 다음에 나오면 그것은 그 URL의 하위 자원으로 여겨진다. Xptr은 그 자체로는 현재문서에 적용된다고 여겨진다.

하나의 Xptr에서는 이미 #, ?, | 중 하나가 먼저 나온다. #과 ?의 의미는 HTML 어플리케이션의 경우와 동일하다. |는 하위자원이 Xptr을 그 자원에 적용하는 것에 의해 발견된다는 것을 의미한다. 그러나 이러한 방법은 어플리케이션의 역할이다.

"[http://etext.virginia.edu/bin/tei-tocs?div=DIV2&id=SAXR" target="new"-TEI Extended Pointer Notation\(EPN\)](http://etext.virginia.edu/bin/tei-tocs?div=DIV2&id=SAXR)은 일부 URL의 마지막에 나오는 '부분 주소(fragment address)'보다 훨씬 강력하다. -href="#FAQ-HYPertext">예를 들어 두 문단 앞의 'Xptr'이라는 단어는 [http://www.ucc.ie/xml/faq.sgm#ID\(faq-hypertext\)CHILD\(2,*\)\(4,*\)](http://www.ucc.ie/xml/faq.sgm#ID(faq-hypertext)CHILD(2,*)(4,*))처럼 적용될 수 있다. 이것은 faq-hypertext라는 ID를 가진 요소의 두번째 child object 내의 네번째 child object라는 의미이다. "[faq.sgm](#)"-

SGML 버전에서 지금 보고 있는 질문--`faq-hypertext'라는 ID를 가지고 있는--의 처음부터 object를 세어 보아라.

1. 문제의 제목

```
<SECT2 ID="faq-hypertext">
<TITLE>How will XML affect my document links?</TITLE>
```

2. 두번째 문단

1. 문단의 시작부터 마크업의 첫번째 아이템까지의 문자 데이터

```
<PARA>The
```

2. 마크업 아이템

```
<ULINK URL="http://www.w3.org/TR/WD-xml-link">XML Linking
Specification (XLL)</ULINK>
```

3. 문자 데이터의 다음 부분

```
document contains a detailed specification. An XML link can
be either a URL or a TEI-style Extended Pointer (
```

4. 그 다음 마크업 아이템

```
<LINK LINKEND="tei-link">Xptr</LINK>
```

이 파일을 *Panorama*나 *MultiDoc Pro*로 보면 예제 문장의 앞부분에 있는 밝게 보이는 십자-참조(cross-reference) 버튼을 클릭할 수 있다. 그리고 'Xptr'이 나오는 부분을 포함하여 거기에 링크되어 있는 모든 Extended Pointer Notation의 지점들을 볼 수 있을 것이다(HTML 브라우저는 양방향 링크나 EPN을 지원하지 않으므로 이렇게 하는 것은 의미없는 일이다).

C.11 XML을 사용하여 수식을 표현 할 수 있는가?

당신이 수식을 지원하는 "#FAQ-DOCTYPE"-문서 형식(document type)을 사용한다면 가능하다. 수학을 사용하는 집단들에서 이러한 소프트웨어를 개발하고 있고,"<http://www.w3.org/Math/>" target="new"-W3C에는 **MathML proposal**이 있다. 이것은 순수한 XML 어플리케이션이다. XML 파편들(fragments)을 만드는 것은 오래전에 소멸한 HTML3이나, "<http://www.arboret.org/~silmaril/dtds/html/htmlpro.html>" target="new"-HTML Pro, "<http://www.sil.org/sgml/gen-apps.html#iso12083DTDs>" target="new"-ISO 12083 Math, "<http://www.can.nl/~abbott/OpenMath/>" target="new"-OpenMath같은 제품들이나, 또는 스스로 만든 것들도 가능 할 것이다. SGML에 기반한 간단한 수식을 보여주는 브라우저들은 이미 존재하고 있다. 예를 들어 *Panorama*, *Multidoc Pro*같은 것이 이런 제품이다.

수식의 복잡성은 x_i 과 같은 수식표현들부터, $E = mc^2$ 과 같이 단순한 문장내의 등식, 그리고 다음과 같은 표현 등식과 같이 다를 수 있다.

$$\sum_{i=1}^n (x_i - \pi)^2/n$$

(만약 당신이 HTML 브라우저를 사용하고 있는데, TeX에 상당하는 코드를 읽는 IBM사의 "<http://www.ics.raleigh.ibm.com/ics/techexp.htm>" target="new"-TechExplorer같은 Netscape 수학 플러그인이 설치되어 있지 않다면, 위의 등식은 정확히 보여지지 않을 것이다.)

C.12 XML은 메타데이터를 어떻게 처리하는가?

XML은 자신만의 고유한 마크업 언어를 정의할 수 있도록 하기 때문에, 어떤 형식의 메타데이터-- 예를 들어 "<http://www2.sub.uni-goettingen.de/dc-wf.html>" target="new"-Dublin Core, Warwick Framework, "<http://www.dstc.edu.au/RDU/RDF/>" target="new"-Resource Description Framework (RDF), "<http://www.w3.org/PICS/>" target="new"-Platform for Internet Content Selection (PICS) 와 같은--라도 저장하고 링크할 수 있는 XML의 확장된 하이퍼텍스트적 특징("#FAQ-HYPertext"-링크에 관한 질문을 참조하라)을 완벽히 이용할 수 있다.

XML은 구조이지 어플리케이션이 아니기 때문에 미리 정의된 요소는 없다. 따라서 저자가 어떻게 메타데이터를 응용해야 한다던가, 또는 메타데이터를 사용해야 한다던가 하는 것을 구체화하는 것은 XML의 임무가 아니다. 그러므로 간단한 속성(attribute)부터 완전한 Dublin Core/Warwick Framework 메타데이터 내부 자료까지 어떠한 적합한 방법이라도 사용할 수 있다. 브라우저 회사들 또한 그들의 고유한 구조 제안과 방안을 내어놓을 수도 있다.

C.13 XML에서 자바(JAVA), ACTIVEX 등을 사용할 수 있는가?

이것은 브라우저 메이커가 어떠한 기능을 제공하는가에 달려 있다. XML은 정보를 기술하는 것에 관한 것이다. 즉, XML의 내장된 기능을 위한 스크립트 언어와 랭귀지들은 최종이용자에 의해 정보가 다루어질 수 있도록 하는 소프트웨어이다.

XML은 그 자체로 스크립트 언어를 실행하는 데 요구되는 마크업을 정의하는 방법을 제공한다. 종립적 표준으로서 XML은 스크립트 언어의 사용을 권장하거나 제한하지도 않는다. 또한 어느 하나의 언어만을 선호하지도 않음으로써, 그 영역은 대단히 공개되어 있다. 개발은 계속진행 중이다. XML 관점에서 "<http://www.datachannel.com/ChannelWorld/XML/dev/>" target="new"-자바 API의 표준화에 관한 존 티그(John Tigge)의 제안을 참조하라.

스크립트 랭귀지들은 "<http://www.w3.org/TR/NOTE-XSL-970910>" target="new"-Extensible Style Language(XSL)을 위해 제공된다("#FAQ-STYLE"-스타일시트(stylesheets)에 관한 질문들을 참조하라).

C.14 표현은 어떻게 제어하는가?

스타일시트의 사용은 XML 내부에 함축되어 있다. 몇몇 브라우저들은 <PARA>, 또는 <ITEM>을 포함하는 <LIST> 같은 유명한 요소를 포함하는 단순한 기본 스타일을 제공할 것이다. 그러나 일반적으로 특정 스타일시트는 저자가 레이아웃(layout)을 더 잘 제어할 수 있도록 해 줄 것이다. 그러나 이용자들의 취향에 따라 다양하게 파일이 보여지는 어떤 시스템과도 마찬가지로 저자는 이용자의 시스템 리소스-폰트와 같은-를 알 수 없다. 저자에게는 상당한 주의가 요구된다.

- SGML 문서를 위한 스타일시트 국제 표준은 "<http://www.sil.org/sgml/related.html#dsssl>" target="new"-DSSSL (the Document Style and Semantics Specification Language)("http://www.iso.ch/" target="new"-ISO 10179)이다. DSSSL은 스타일시트와 문서 변환을 위한 스키마와 유사한 언어를 제공하고 있으며, Jade formatter에서 널리 사용되고 있다.
- "<http://www.w3.org/Style/css>" target="new"-Cascading Stylesheet Specification (CSS)은 요소들에 스타일을 부여하는 간단한 구문(syntax)를 제공하며, HTML 브라우저에서 사용되어 왔다.
- Panorama와 MultiDoc Pro에서 이미 사용된 Synex stylesheet DTD.
- 새로운 "<http://www.w3.org/TR/NOTE-XSL-970910>" target="new"-Extensible Style Language (XSL)이 특별히 XML과 사용되기 위해 제안되고 있다. XSL은 XML 구문을 사용하며-실제로 스타일시트는 XML 파일이다-

DSSSL과 HTML의 CSS 양쪽 포맷의 특징을 합친 형태이다. 그리고, 이미 몇몇 중요한 회사 제품에서 지원되고 있다.

이제 어떤 브라우저들이 이것을 사용하지는 지켜보는 것이 남았다.

D. DEVELOPERS AND IMPLEMENTORS (INCLUDING WEBMASTERS AND SERVER OPERATORS)

D.1 스펙을 볼 수 있는 사이트

바로 "<http://www.w3.org/TR/WD-xml>" target="new"-여기(<http://www.w3.org/TR/WD-xml>)에서 구할 수

있다. "<http://www.fxis.co.jp/DMS/sgml/xml/wd-xml-lang.html>" target="new"-일본어 버전은

<http://www.fxis.co.jp/DMS/sgml/xml/wd-xml-lang.html>과 "<http://www.fxis.co.jp/DMS/sgml/xml/wd-xml-link.html>" target="new"-<http://www.fxis.co.jp/DMS/sgml/xml/wd-xml-link.html>에서 볼 수 있다.

D.2 'DTDLESS', 'VALID', 'WELL-FORMED'란 무슨 의미인가?

완전한 SGML은 마크업(elements)을 어떤 특정한 문서 타입에 사용가능한 것으로 기술하기 위해 DTD를 사용한다. 하지만 DTD의 설계와 구성은 복잡하고 간단치 않은 일이다. 그래서 XML은 DTD 없이도 사용될 수 있게끔 설계되어 왔다. 'DTDless 작동'이란 마크업을 정해진 방식으로 정의하지 않고 새로 만들 수 있음을 의미한다.

실제로 DTDless 파일은 비공식적 형태로 자신의 고유한 마크업을 요소들(elements)의 존재와 위치를 이용해 '정의한다'. 그러나 브라우저와 같은 XML 어플리케이션이 DTDless 파일을 접하게 되는 경우, 브라우저는 문서구조를 이해할 수 있어야 한다. 왜냐하면 DTDless 파일은 브라우저에게 도움을 줄만한 아무런 DTD를 갖고 있지 않기 때문이다. 따라서 이 문제를 해결할 몇몇 보완작업들이 진행돼 왔다.

위에서 제기된 문제의 예를 들어 보자. HTML의 요소는 'EMPTY'로 정의되는데, 이는 엔드태그(end-tag)를 가지고 있지 않다. DTD가 없으면 XML 어플리케이션은 어떤 요소에 엔드태그를 붙여야 할지 말아야 할지를 판단할 수 없게 된다. 그래서 'well-formed'란 개념의 도입이 필요하게 되었다. 'well-formed'란 모든 요소의 시작과 끝을 만드는 것이다. 'well-formed' 개념에 비추어보면 EMPTY 요소의 발생은 전혀 모호한 문제가 아니다.

D.2.1 'WELL-FORMED' 문서들

모든 XML 문서들은 정확한 형식을 갖추어야 한다.

ame="FAQ-RMD"-

§ 사용하고 있는 DTD가 없다면 XML 문서는 다음과 같은 "#FAQ-RMD"-Standalone Document Declaration(SDD)로 시작한다.

```
<?XML version="1.0" standalone="yes"?>
<foo>
  <bar>...<blort/>...</bar>
</foo>
```

- 모든 태그들(tags)은 쌍을 이루어야 한다. 즉, 문자 데이터를 담고 있는 모든 요소들(elements)은 시작 태그(start-tag)와 마침 태그(end-tag)를 모두 가지고 있어야 한다. (독립 요소들("#FAQ-EMPTY"-empty elements)을 제외하고는 예외는 있을 수 없다. 아래를 보시오.);

- 모든 속성 값(attribute values)은 인용부호 안에 있어야 한다 (작은 따옴표(apostrophe)는 속성값이 큰 따옴표(")를 포함하고 있을 때 사용할 수 있다. 그리고 그 반대의 경우도 마찬가지이다.). 만약 둘 다 필요하다면 '와 "을 사용하여야 한다.
- HTML의 , <HR>,
 등과 같은 마침 태그가 없는 독립 요소의 태그들(EMPTY element tags)은 반드시 '/'로 끝내거나 실제 마침 태그(end-tag)를 붙여 비어있지 않은(non-EMPTY) 요소로 만들어 주어야 한다.
예 :
은
이나
</BR>이 되어야 한다.
- § 텍스트 데이터에 <나 &와 같은 마크업에 사용되는 문자를 사용하여서는 안된다 (다시말해 그것들은 <와 &로 사용하여야 한다.).]]의 경우는 CDATA섹션으로 선언된 경우가 아니라면]]>로 표현한다.
- 요소와 요소는 각각의 내부에 적절히 중첩되어야 한다 (SGML에서와 마찬가지로 오버래핑(overlapping) 마크업은 불가능하다).
- DTD가 없는 Well-formed 파일들은 속성들(attributes)을 어떤 요소(element)에도 사용할 수 있지만, 속성들은 초기값 지정에 의해 CDATA 타입이어야 한다.

DTD가 없는 Well-formed XML 파일은 <, >, ', ", & 등을 가지고 있는 것으로 미리 정의되었기 때문에 DTD없이도 이것들의 사용이 가능하다. 그러나 Valid XML 파일들은 이것들을 사용하려면 명백히 선언을 하여야 한다.

D.2.2 VALID XML

"#FAQ-DOCDEC"-Valid XML 파일들은 다른 모든 "#FAQ-SGML"-SGML 어플리케이션과 마찬가지로 "#FAQ-DTD"-Document Type Definition (DTD)를 가지고 있으며, 또한 DTD에 충실하다. Valid XML 파일들은 당연히 "#FAQ-WF"-well-formed 하여야 한다.

Valid 파일은 다른 SGML 파일들과 마찬가지로 DTD로 시작하지만, 선택적으로 XML 선언(XML Declaration)을 가질 수도 있다.

```
<?XML version="1.0"?>
<!DOCTYPE advert SYSTEM "http://www.foo.org/ad.dtd">
<advert>
  <headline>...<pic/>...</headline>
  <text>...</text>
</advert>
```

"#FAQ-SPEC"-XML 스펙은 모든 인스턴스에 적용되는 "http://www.ucc.ie/xml/sgmlxml.decl" target="new"-XML을 위한 SGML Declaration을 정의하고 있다. 특정한 DTD의 "#FAQ-DTD"-XML 버전은 로컬에서 이용가능한 형태로든--다시말해 이용자가 이미 디스크에 복사본을 가지고 있는 경우--, 아니면 네트워크를 통해 검색가능한 형태로든 XML 프로세서에 접근할 수 있어야 한다. 이것은 위와 예와 같이 System Identifier에 DTD의 URL을 입력함으로써 가능하다. 몇몇 사람들이 선호하는 것처럼 "http://www.ucc.ie/cgi-bin/public" target="new"-Formal Public Identifier를 지원하는 것도 가능하다. 하지만 그럴 경우에도 FPI는 System Identifier보다 선행하는 것일 뿐, 시는 여전히 주어져야 한다.

```
<!DOCTYPE advert PUBLIC "-//Foo, Inc//DTD Advertisements//EN" "http://www.foo.org/ad.dtd">
```

XML 선언의 다른 속성들을 위한 초기값들은 VERSION="1.0" and ENCODING="UTF-8"이다.

D.3 SGML과 XML 사이의 변화는 이밖에 무엇이 있는가?

중요한 변화들은 DTD를 작성에 있어 무엇을 할 수 있는가와 관련되어 있다. 구문을 단순화하고, 처리 소프트웨어를 작성하는 것을 쉽게하기 위해, 많은 마크업 정의 옵션들이 생략되었다("#FAQ-SPEC"-XML 스펙의 부록 A를 참조하라).

새로운 구획문자(delimiter)인 콜론(:)의 사용이 네임스페이스(namespaces) 실험을 위해 이름(Names)에 허용되었다-이것은 DTD가 요소 소스, 소유권, 어플리케이션을 구별할 수 있도록 해 준다. 콜론(:)은 시작이나 끝이 아닌 이름중간(mid-name)에서 사용될 것이며, 이 구문은 다음 버전에서 변경될 것이다.

D.4 오늘날 어떠한 XML 소프트웨어가 사용가능한가?

¶ 이 FAQ에 언급되기에는 너무나 빠른 변화 때문에 자세한 사항은 언급하지 않겠다. "<http://www.sil.org/sgml/xml.html>" target="new"-<http://www.sil.org/sgml/xml.html>에 있는 XML 페이지를 참조하라.

브라우저에 관해서는 "#FAQ-BROWSER"-XML 브라우저에 관한 질문을 참조하고, 소프트웨어 개발자들은 "#FAQ-MAILINGLIST"-xml-dev 메일링리스트의 상세정보를 참조하라. 버트 보스(Bert Bos)가 bison, flex, perl, Python 에 관련된 "<http://www.w3.org/XML/notes.html>" target="new"-XML 개발 리스트를 제공하고 있다.

D.5 XML을 사용하기 위해 현재의 서버 소프트웨어를 수정해야 하는가?

.xml 파일을 올바른 MIME 형식으로 지원하기만 하면 된다. text/xml과 text/xsl의 MIME 형식이 사용가능하므로 XML을 지원하기 위해 필요한 모든 것은 mime-types 파일과 이것에 대응하는 것들을 다음을 추가하여 수정하면 된다.

text/xml xml XML

text/xsl xsl XSL

XML은 스타일쉬트와 정교한 하이퍼링크를 지원하도록 설계되었기 때문에 XML 문서들은 DTD, 엔터티 파일, 카탈로그, 스타일쉬트와 같은 보조파일들을 동반할 것이다. 이러한 파일들은 고유의 MIME 항목(entry)를 요구하며, 적절한 디렉토리 안에 놓여야 한다. XML 워킹그룹이 계획하고 있는 것들 중 하나인 XUA(XML User Agent)는 하나의 메시지에 XML 문서와 XSL 스타일을 묶는 기법을 제공할 것이다.

만약 HTML을 생성하는 스크립트를 사용하고, 이것이 XML과 같이 기능하기를 원한다면, 적절한 문서 형식을 생성하도록 수정되어야 할 것이다.

D.6 SERVER-SIDE INCLUDE를 계속 사용할 수 있는가?

그렇다. 그것들이 결과물을 XML을 준수하는 파일(XML-conformant file), 즉 "#FAQ-VALID"-valid 또는 단지 "#FAQ-WF"-well-formed한 파일의 일부분으로 생성한다면 가능하다.

D.7 CLIENT-SIDE INCLUDE를 계속 사용할 수 있는가?

"#FAQ-SSINCLUDES"-server-side INCLUDE와 같은 규칙이 적용된다. 예를

들어 SDQL 질의어, Java 언어, LiveWire 질의어나 스트림 데이터 등의 써드 파티 엔진(third-party engine)에 넘겨지는 어떠한 내부 코드(embedded code)도 XML 마크업으로 잘못해석될 가능성있는 문자를 포함하지 않는지를 확인해야 한다. 다시 말해 <, >, &를 포함하고 있는지를 확인해야 한다. XML 어플리케이션이 내부 코드를 파싱하는 것을 피하기 위해 CDATA라고 표시를 하든지, 표준방식인 <, >, &를 사용한다.

D.8 나는 XML 스펙을 이해하고자 노력하고 있다: 왜 SGML(과 XML)은 그렇게 어려운 용어를 사용하는가?
구현(implementation)이 성공하기 위해서는 용어(terminology)가 정확해야 한다.

예를 들어 보자. '요소(element)'와 '태그(tag)'는 동의어가 아니다. 요소는 자신의 마크업을 갖는 하나의 완전한 정보 단위이며, HTML에서
과 같이 시작 태그 만으로 구성될 수도 있고, 시작태그-마침태그-그 사이의 내용으로 구성될 수도 있다. 태그는 그 자체만으로는 요소의 시작과 끝을 표현할 뿐이다.

스펙에 있어서 두루뭉실한 용어들은 오해를 유발한다. 따라서 공식적인 표준은 공식적 용어로 서술되어야 한다. 이것은 공식 문서가 아니다. 따라서 명철한 독자들은 이미 '요소이름(element names)'이라는 용어가 '요소타입이름(element type names)'이라는 보다 정확한 용어를 지칭함을 눈치챈 것이다. '요소타입이름'이 보다 정확한 용어이지만 '요소이름'이 보다 폭넓게 사용되고 있는 형편이다.

"#FAQ-SGML"-SGML을 처음 접하는 사람들은 "<http://www-tei.uic.edu/orgs/tei/>" target="new"-

TEI의 "<http://etext.virginia.edu/bin/tei-tocs?div=DIV1&id=SG>" target="new"-Gentle Introduction to SGML 등과 같은 글을 읽고 싶어할 것이다.