



EDIFECs

THE DNA OF B2B

*E-business XML
Document Style Guide
Version 2
27 October 2000*

Contents

Document Executive Summary	i
Preface	ii
1 Introduction	1
2 General Principles	2
3 Comments	3
4 Code Layout	4
4.1 Angle Brackets and Parenthesis	4
4.2 Declarations	5
4.3 Spaces	5
4.4 Wrapping	5
5 Naming Conventions	6
5.1 Files and Directories	6
6 Namespaces	7

The information contained in this document represents the current views of Edifecs on the issues discussed as of the date of publication. Because Edifecs must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Edifecs, and Edifecs cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. EDIFecs MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Product and company names mentioned herein may be the trademarks of their respective owners, Edifecs, CommerceDesk, and SpecBuilder are trademarks of Edifecs.

©2000 Edifecs.

Document Executive Summary

The eXtensible Markup Language (XML) is becoming the preferred syntax of e-business messages. XML is a W3C¹ standard that allows language designers to construct e-business grammars and e-business documents. This document specifies the grammatical style used to specify the grammar and document structure for XML messages that are compliant with the Business Collaboration Framework (BCF).

¹ <http://www.w3c.org/>

Preface

Partners exchange business data as documents whose grammar and structure are defined using the eXtensible Markup Language (XML). There are a number of reasons for using XML over other data definition languages.

- The XML permits the construction of self-describing data, e.g. `<Price>20</Price>`.
- The XML is agnostic to the particular source and target computer programs and platforms.
- The XML is a technology well supported by solutions.

XML provides tremendous flexibility in language design. This can at first seem very powerful but very quickly demands some rules for consistency. Consistency makes XML specifications easier to write and read because after—all humans still need to craft and maintain the language. Accordingly, specific style guidance is needed.

Purpose of the Document

The purpose of this document is to provide the specific grammatical style convention that should be applied consistently in e-business collaboration message design.

Intended Audience

XML designers, programmers and testers will find that the conventions found in this document will make their work easier to integrate with the efforts of other efforts that use the Business Collaboration Framework (BCF). It will be easier to understand, and more importantly, easier to integrate into automated and semi-automated system specification development processes.

Prerequisites

It is assumed that the audience will be familiar with or have knowledge of the following technologies and techniques:

- The XML standard

Scope of the Document

The principle function of this document is to establish a consistent standard that will provide for easier maintenance of code. This will benefit the team and the project in that those who are new to the code can quickly orient themselves, and thereby sooner become productive members of the team. It is intended to be a dynamic document and can be reviewed as needed.

Style Conventions

This document uses typographical and language conventions to convey specific meanings.

Typographical Conventions

The use of a *bold/italic font* indicates a UML or business process metamodel entity name.

Language Conventions

This specification adopts the conventions expressed in the IETF's² RFC 2119 "Key Words for Use in RFCs to Indicate Requirement Levels." The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Authors

Arthur Greef, Gary Ham, Jim Clark, John Yunker, Mark Smith, and Tony Weida.

Acknowledgments

Edifecs³: Edifecs is administering the creation of the Business Collaboration Framework (BCF). The BCF is a collection of documents that prescribe the policy, architecture and specifications for executing business collaborations for e-business.

- For general questions send mail to: support@edifecs.com

Contacts

Jim Clark—Edifecs Business Collaboration Services (BCS) Director of Industry Solutions

JamesC@edifecs.com

Tony Weida—Edifecs BCS Enterprise Architect

weida@edifecs.com

Copyright

© 2000 Edifecs. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

² <http://www.ietf.org/>

³ <http://www.edifecs.com/>

1 Introduction

The following document sections contain standards, guidelines, and rationales. Guidelines must be adhered to unless there is compelling reason to deviate. Deviation from a guideline must be discussed and approved during the code walk-through. A standard is an item to which compliance is mandatory. Deviation from a standard must be discussed, approved, and signed-off on by the team lead during the code walk-through. Rationales have been used where necessary to explain the meaning of an item, or why it was chosen.

- General Principles—This section contains the basic philosophy a developer should keep in mind while coding.
- Comments—This section deals with the placement and contents of comments in the code.
- Code Layout—This section has to do with the alignment of the code, white space, declarations and keywords, and where they should all be located.
- Naming Conventions—This section contains the structure for naming XML element, attributes, entities and files.
- Usage—This section concerns the “how” and “when” certain constructs should be used (for example, loops, inheritance, error handling, etc.)
- Document Layout—This section applies to where things should be located in DTDs and XML documents.
- Coding Examples—This section contains an example of a header file and module that conform to this XML Coding Standard.

2 General Principles

The primary goal of the coding standard is maintainability. Other important considerations that relate to the spirit of the standard include correctness, readability, consistency, extensibility, portability, clarity, and simplicity. When in doubt, the programmer should strive for clarity first, then efficiency.

Think of the reader. Do not just write for yourself. Keep it simple. Break down complexity into simpler chunks. Clearly comment necessary complexity. Be explicit. Avoid implicit or obscure language features. Be consistent. Minimize scope, both logical and visual.

3 Comments

Guideline: Be clear and concise. Say what is happening and why. Do not restate code.

Guideline: Keep code and comments visually separate.

Standard: Use header comments for all files.

Example: <!--

```
SpecBuilder Properties
```

```
Version 1.0
```

```
Copyright 2000 Edifecs.
```

```
http://www.edifecs.com
```

```
http://www.commercedesk.com
```

```
-->
```

Standard: Block comments are at the same indentation level as the block they describe.

Standard: Comments more than one line long must have the comment initiating syntax "<!--" on the first line, a following blank line, the comment on multiple lines, another blank line and then the comment terminator "-->".

Guideline: Single line comments may have initiating and terminating syntax on the same line as the comment. In these cases, a space must be provided between the comment syntax and the comment.

Example: <!-- Comment -->

Standard: Ensure comments are correct (and stay correct).

Standard: Do not use trailing comments.

4 Code Layout

Guideline: Use white space wisely.

Guideline: Use block comments to separate the chunks.

Standard: There must be no vertical space between an element declaration and its corresponding attribute list declaration.

Example:

```
' Property ( #PCDATA ) >
    <!ATTLIST Property
        attribute CDATA #IMPLIED >
```

Standard: Indentation level is four spaces.

Standard: There must be one vertical space preceding an element declaration.

Standard: Align the declarations in the body of an element declaration for clarity.

Example:

```
<!ELEMENT Foo ( Bar,
                Bar1 ) >
```

Standard: Align the declarations in the body of an attribute list declaration for clarity.

Example:

```
<!ATTLIST Property
        attribute CDATA #IMPLIED
        attribute2 CDATA #IMPLIED>
```

4.1 Angle Brackets and Parenthesis

Guideline: Use parenthesis to group within a declaration and to emphasize evaluation order in a long expression, even if the order is determined correctly by precedence.

Guideline: Avoid unnecessary parentheses, while keeping it readable.

Guideline: Avoid deep (more than about three) levels of parenthesis nesting.

4.2 Declarations

Standard: Start each declaration on a new line.

Guideline: Use vertical alignment to ease scanning of declarations.

Standard: Do not declare entities until they are used.

Standard: The body of a collection element must not be #PCDATA as this implies that there is implicit punctuation that is not checked by an XML parser. The collection should also not be empty.

4.3 Spaces

Standard: There should be one space preceding a terminating angle bracket ("`>`").

Standard: There should be one space following an XML DTD keyword and a generic identifier (GI).

Standard: Do not space before separators (comma) but do space the other side.

Standard: Balance spacing inside parenthesis.

Standard: Use blank lines before and after block comments.

Guideline: Use vertical alignment to indicate association.

Standard: Use spaces, not tabs.

Rationale: Tab sizes vary on different systems. When spaces are used, the alignment is maintained no matter where the file is edited.

Standard: Place a space between keywords and the following parenthesis.

4.4 Wrapping

Standard: No line of code shall extend beyond column 100. It is acceptable to wrap prior to column 100 however.

Standard: When wrapping lines, indent the continuation line past the current indent column.

5 Naming Conventions

- Standard:** Spell words using correct American English spelling. For the most part avoid abbreviations.
- Guideline:** Make names clearly unique. Avoid similar-sounding names and similarly spelled names.
- Standard:** "Class" element names must start with a capital letter and then use mixed case e.g. "Segment."
- Standard:** "Property" element names must start with a lowercase letter and then use mixed case, e.g. "minimumLength," "maximumLength."
- Standard:** Attribute names must start with a lowercase letter and then use mixed case.
- Standard:** Be consistent when capitalizing abbreviations—make all letters capital or all letters lowercase.
- Standard:** Namespace designations must always be lowercase.
- Standard:** Use mixed case to distinguish name segments instead of underscores.
- Standard:** Use capital letters to begin new name segments within the name.
- Standard:** Name collections using plurals.
- Standard:** Name objects using singular names.
- Standard:** Entity names must start with a lowercase letter and then use mixed case.
- Guideline:** Make ID literals all capital letters.

5.1 Files and Directories

- Guideline:** Name files like element names. Long element names are encouraged.

6 Namespaces

Standard: All namespaces defined by Edifecs must be located on "<http://www.commercedesk.com/>."