1	ebXML Core Component Project Team
2	3/Jul/2000
3	
4	
5	
6	
7 8	
8 9	
10	
11	
12	
13	
14	
15	
16	
17 18	
18	
20	
21	
	Methodology for describing Core Components
22	
23	Analysis and Proposal
24	(Draft-Rev.04)
25	
26	
27	
28	
29	
30	
31 32	
33	
34	
35	
36	
37	
38	
39 40	
40 41	
42	
43	
44	
45	
46	
47	
48	By Work Crown 7 of Core Comparent Project Team
49 50	By Work Group 7 of Core Component Project Team
50 51	Hisanao Sugamata
52	Martin Bryan

53	Contents
54	
55	
56	1. Introduction
57	
58	2. Business entity analysis
59	2.1 Example <party></party>
60	2.2 Requirements for defining the 'Party' entity within
61	UN/EDIFACT
62	
63	3. Common Business Entity concept
64	3.1 Representation Class candidates
65	3.2 Core Component Meta model
66	1
67	4. Considerations
68	4.1 Reusability
69	4.2 Syntactically neutral
70	4.3 Basic Semantic Register
71	4.4 Naming the component
72	4.5 Segmentation of Core Components
73	
74	5. Describe the Core Component
75	
76	6. Sample description
77	
78	7. Instructions for capturing ebXML Core Component definitions
79	7.1 Defining a Pattern
80	7.2 Defining an Entity
81	7.3 Defining a Data Representation
82	7.4 Defining a Data Format
83	7.5 Defining a Code Set
84	
85	Appendix : Core Component Templates

# 8687 1. Introduction

88

94

96

97

98

99

100

101

102

106

110

111

112

113

The first step for achieving to get specifications of the ebXML core components will be analysis of the existing data models currently used in e-business. The concept model for describing the business entities, which were nominated as the candidates of the core components at the  $2^d$  project team meeting of ebXML in Orlando, will be proposed in the paper. This paper will also offer the templates for describing the results of analysis on the business entities.

95 The purposes of the paper includes,

- 1. To show the way to OO-edi methodology through the reverse engineering for the current EDI messages for business experts, who have been engaged in designing legacy EDI messages; and
  - 2. To show how to analyze business entities currently being used in EDI in order to design the object classes for object modeling experts, who have been engaged in designing the software in the manner of Object Oriented Approach (OOA).

103 It is expected that business entities, which are selected and analyzed by the business experts of the 104 Core Component Working Group, can be used in the business process models. The entities can also 105 be stored in the repository(s) aligning with the ebXML standards.

107 The analysis and proposal prepared in this paper, respecting the ebXML requirements specification,108 is intended to be,

- 109 1. Syntactically neutral,
  - 2. Conforming to ISO11179 and
  - 3. Aligning with the Unified Modeling Methodology.
  - 4. A mechanization for developing Core Components

#### 114 **Revision note :**

115 There have been several discussions around the first draft of this paper.

- 116 One of the disputing subjects was how to describe the analysis pattern for the business entities.
- 117 You can find the analysis and the proposal for describing the pattern in the section 4.5.

118 In the  $3^{rd}$  revision of the draft paper, Representation Class candidates are introduced in the section 3.1.

- 120 In addition to the previous version of the paper, Mr. Martin Bryan proposed the XML based 121 templates instead of the paper-based form for mechanizing the development of Core Components.
- 122 In the appendix file attached you can find the usable templates in XML based form.
- 123 In the 4<sup>th</sup> revision of the draft paper, the XML based templates are amended through some testing.
- 124 Also the meta model for Core Component is added in the section 3.2.

#### 2. Business entity analysis

At the 2<sup>nd</sup> meeting of the core component team in Orlando, typical business entities were on the table for discussion. One of of the business entities discussed was 'Party'.

#### 2.1 Example <Party>

The business entity 'Party' is observed in several patterns or within contexts. The 'Party' is defined within segment groups in the UN/EDIFACT messages. 

In the UN/EDIFACT messages of ORDERS and INVOIC, the pattern of the business entity 'Party' has rich attributes which is specially used in purchasing applications.

194	nas ne	in autobutes which is specially used in purchasing applications.
135 136		Dentry
136		Party
137		AD : Names and addresses of the parties relevant to the order/invoice.
139		The qualifier of NAD is specifying the function of the party (seller or buyer)
140		LOC : Specific location information of the party
140		Loe . Speene location information of the party
142		FII : Financial institution and relevant account number for the party
143		I II . I mailetal institution and relovant account number for the party
144		RFF : Reference for the party
145		
146		DTM : Date and/or time related to the reference
147		
148		DOC : Information relating to the documents required by the party specification
149		
150		DTM : Date and/or time related to the document
151		
152		CTA : Person or department whom communications should be directed
153		
154		COM : Communication type and number for the contact
155		
156		
157		Fig.1 Party example – 1
158		
159		e other hand, the 'Party' entity in other message types have a simpler pattern. The simple
160		is can be seen in the messages, such as CUSDEC (Customs declaration message), DELFOR
161		ery schedule message), BANSTA (Banking status message), BUSCRD (Business credit report
162		ge) and IPPOAD (Insurance policy administration message). The typical pattern of the simple
$\begin{array}{c} 163 \\ 164 \end{array}$	form o	f 'Party' is as follows.
164 165	ſ	Destry
165		<b>Party</b> (PNA : Name of the party and their function relevant to the message.
167		1 INA . Ivalle of the party and then function relevant to the message.
167		ADR : Addresses of the party.
169		The reaction of the party.
170		CTA : Person or department whom communications should be directed
171		errere erson of department whom communications should be uncered
172		COM : Communication type and number for the contact
1		

Fig.2 Party example -2

\*Note: RFF segment is also used in several messages in addition to the above pattern.

The above examples of the pattern for 'Party' are representing the organizations involved in the relevant business. Sometimes the one side of the business party may be an individual person. The
single person case can be seen in the application of the medical industry, the life insurance industry
or the labor market industry. The following sample shows the pattern in the message MEDREQ
(Medical service request message)

(ivieui	cal service request message)
·	Dentry
	<b>Party</b> (PNA : Name of the party or professionals and their function relevant to the message.
	ADR : Addresses of the healthcare party.
	COM : Communication number for healthcare party.
	FTX : Unstructured telecommunication numbers of the party (ex. Email)
	RFF : Alternative identification number assigned to the party
	SEQ : Allocated sequence number to the party. LAN : The language used by the healthcare party.
	SPR : The medical specialty of a healthcare organisation.
	QUA : The professional medical qualification.
	EMP : The position or rank of a healthcare professional.
	Ewi . The position of rank of a headilicate professional.
	Fig.3 Party example – 3
	rig.5 raity example = 5
In the	above example, the party can be the organization or the individual professional person.
	the entity 'Party' is used for the individual, another entity 'Person' is recommended. Using th
	'Person' avoids the complex pattern of the entity.
entity	reison avoids the complex pattern of the entity.
We sh	ould consider which pattern is efficient for being used in the e-business. There may be sever
	rements to decide which is better for ebXML standard.
measu	tements to decide which is better for control standard.
Two n	neasurements shall be considered as the first priority. The measurements are (1) 'Reusability i
	1 applications' and (2) 'Usability in the specific application'. We should carefully model th
	ss entity patterns considering the tradeoff between two of them. (see Section.4)
In abo	ve samples of the business entity 'Party', we can choose the second one (Fig.2) as one of the
	on business entities. We can choose it because it is used in various kinds of applications.
	TI TI
2.2 Re	quirement for defining the 'Party' entity within UN/EDIFACT
(1) At	ostraction
Th	e first segment of the 'Party' entity has a qualifier, which is specifying the function of the
par	rty.
Ťŀ	he first data element 3055 specifies the function of the party. (see Fig.4)
	e party without the qualification has no meaning in the real business. The party can be 'th
	ler', 'the buyer' or the other party who has the specific function in the relevant business.
	other words, the entity 'Party' is an abstraction for many roles in the business. We can call the
	arty' as a super class and the party who has the specific function in the business as a sub-cla
	the object-oriented world. In the XML world the party would have meaning based upon the

- 223 context of where the party element was included in the hierarchical model.
- 224
- 225 
  225 
  225 A section 4.5.

227	
228	PNA : Party identification
229	3035 Party function code qualifier
230	C082 Party identification details
230	3039 Party identifier
231	1131 Code list identification code
232	
233 234	3055 Code list responsible agency code
	*Note: Above structure is selected for the organizational
235	party used in trading.
236	
237	
238	Fig.4 PNA segment structure
239	
240	(2) Relation
241	The entity 'Party' is composed with the segments PNA(Party identification), ADR(Address),
242	CTA(Contact information), COM(Communication contact) in UN/EDIFACT messages. PNA
243	(Party identification) is used only in the entity 'Party', but others can be used in other relations of
244	other entities. For example, ADR (Address) may be used for specifying the delivery address.
245	Therefore, it is clever to distinguish the entity and the relation of the entities (the pattern). The
246	entities and the patterns are the candidates for the core components of ebXML standards.
247	
248	(3)Attributes
249	In the Fig.5, you can see several attributes for the entity 'Party'.
250	
251	Party
252	Attribute-1 Party identification $\rightarrow$ a code value
253	Attribute-2 Address
254	Attribute-3 Contact
255	
256	
257	Fig.5 Attributes of Party
258	
259	The first attribute 'Party identification' has a value that identify the party. The second and third
260	attributes shall be specified through the other entities. The address of the party is specified by the
261	attribute of the entity ADR. The telephone number or E-mail address for contacting the Party is
262	specified through the entity CTA and COM.
263	
264	(4) Representation
265	When the attributes get their values, the characteristics for them shall be defined. The
266	identification of Party may be coded form, the address may be specified in Postal form, the
267	telephone number may be numeric and the E-mail address may be character string. Any values of
268	attribute shall have their certain representations.
269	
270	(5) Value
271	The first data element (3039) of the composite data element C082 in Fig.4 shall
272	have the value 'Party identifier'. The value of 'Party identifier' shall be specified in a
273	code list that is defined by the following two data elements. There can be many code
274	lists. Even the same value of the data element 3039 has deferent meaning in the
275	deferent code list. These code lists are called value domains.
276	When the data element has the number representation, the value can be any
277	arithmetical value. All the arithmetical values are one of the value domains.

When the data element has the calendar date, the value can be specified in the Gregorian calendar dates. All the calendar dates are another value domain.

#### 280 **3. Common Business Entity concept**

Through the analysis of the business entities and the patterns in the previous section, the concept model for Common Business Entity can be described as follows.

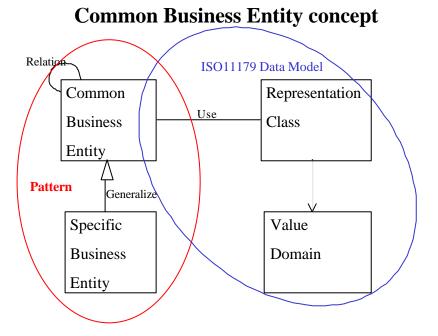


Fig.6 Common Business Entity concept

- (1) Common Business Entity is a concrete class or an abstraction class generalized from one or more Specific Business Entities.
  - (2) A concrete class of Common Business Entity or a Specific Business Entity has one or more instances.
- (3) Common Business Entity may have any kind of relations to other Common Business Entities. The series of relations related to a certain business behaviour is called Pattern.
  - (4) Common Business Entity shall have attributes those are related to other Common Business Entities or shall use Representation classes.
- (5) Representation classes have properties. The properties for representation classes have a value domain, data type and, if
  - necessary, a unit of measure or a character set.
  - (6) The concept (lexical meaning) and the format (syntactical expression) are the two schemas of a data type.
- (7) The value domain is defined in the scope of certain concept (lexical meaning)with the certain format (syntactical expression).
- 301

283

284 285

286 287

288

289

290

291 292

293

294 295

296 297

- 302 The following items are kinds of Core Components.
- 303 (1) Analysis Pattern
- 304 (2) Common Business Entity
- 305 (3) Specific Business Entity
- 306 (4) Representation class
- 307 (5) Value domain
- 308
- 309 Analysis patterns may be used in the business process.
- 310 Analysis patterns, Common Business Entities and Representation classes shall be 311 registered in a ebXML compliant repository.
- 312 The highly reusable Specific Business Entities shall be registered in a ebXML 313 compliant repository. Other ones may be defined by each specific application for each 314 specific industry.
- The values in the highly reusable value domains shall be registered in a ebXML compliant repository. The application unique value domains, including code sets, may be defined by each specific application for each specific industry.
- 318

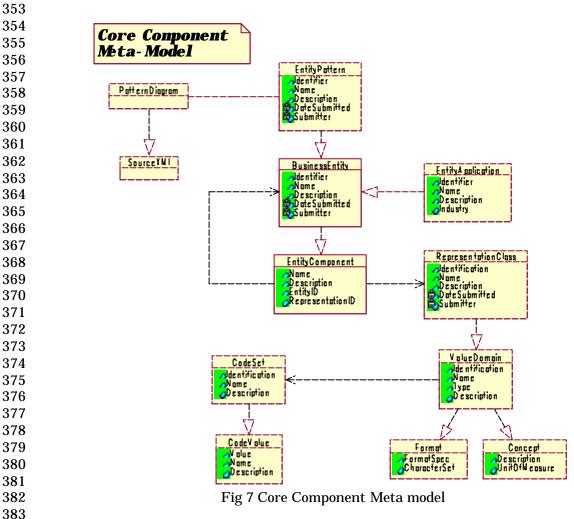
# 319 3.1 Representation Class candidates320

321 The following lists are the candidates for Representation Class.

- A number of monetary units. It is normally associated with a 322 amount 323 type of currency. A character string that represents a member of a set of values. 324 code 325 description A series of sentences describing a person, object, place, event 326 or concept. 327 identifier A character string used to identify and distinguish uniquely, one instance of a value within an identification scheme. 328 329 name A word or phrase that constitutes the distinctive designation 330 of a person, object, place, event or concept. What the person, object, place, event or concept is known by or called. 331 number 332 An arithmetical expression representing a particular value. 333 Note: This may often be used to imply sequence or a member 334 of a series. A rate expressed in hundredths between values that have the 335 percent same unit of measure. 336 A number of non-monetary units. It is normally associated 337 quantity 338 with a unit of measurement. A quantity or amount measured with respect to another 339 rate 340 measured quantity or amount. A date and/or time as measured in the time dimension. 341 date or time A length of time that a person or thing has existed. 342 age 343 Indicator An attribute indicating a condition such as on/off, true/fouls, yes/no, 0/1. 344 A standard unit used to express size, amount or degree.(\*) measure (\*) Another idea is to use 'height', 'width', 'length', 'degree' or 'size' instead of 'measure'. 345
- 346 347
- **3.2 Core Component Meta model**

348	-
349	The figure 7 shows the meta model for Core Component.

350 351



## 384 4. Considerations

385

## 386 **4.1 Reusability**

387 The reusability of the core component is the key concern for efficiency of the e-business 388 application software development. Reusing the standardized core components can push 389 forward the interoperability between the e-business applications. On the other hand, 390 having too much attachment to the widelevel usability in any application, you may recognize that only atomic level objects can be standardized, such as Numeric, 391 Monetary amount, Gregorian calendar or Percentage. These atomic level objects are 392 393 highly reusable. However, any other structured objects or object patterns have to be 394 developed when you implement any e-business application. It makes it very difficult to 395 implement the interoperability of applications.

396 When highly constructed objects are agreed upon for interchange, you can minimize 397 your development efforts for implementation of the e-business application. The turnkey 398 application packages or the fixed format EDI messages are easy to implement if all the parties involved are using the same platform (hardware) and the same application 399 400 package (software). But it is impossible for a single uniformed application to be installed in all the enterprises in various industries. Therefore, the high level 401 402 constructed components can be used only in certain applications in certain industries, 403 or can be used only between certain trading partners.

404 Because of the above considerations, we shall select and standardize the proper level of

- 405 components.
- 406

## 407 **4.2 Syntactically neutral**

408 It is true that no model can be described without any syntax. This may be one of FDTs (Formal 409 Descriptive Technique), natural languages or graphical charts. The meaning of a syntactically neutral 410 model is a model free from an implementation level syntax, such as EDIFACT, XML or JAVA. In 411 other words, the model described in a syntactically neutral manner can be implemented in the 412 computer system using any proper implementation level syntax. Also the syntactically neutral model 413 described by the certain FDT for modeling can be mapped to another model using another FDT.

- 414 UML (Unified Modeling Language) may be one of the FDTs used for the syntactical neutral 415 modeling.
- 416

### 417 4.3 Basic Semantic Register (BSR)

According to the definition of TC154-BSR project, BSC (BSR Semantic Component) is a generic term comprising the components of BSR semantic units. In the context of the BSR there are two types of BSR semantic component, representation class and concept. The concept of BSC is almost same as Common Business Entity of Core Components, and the representation class of BSR is same as Representation class of Core Components. When Common Business Entities and Representation classes are selected and specified, the work done by TC154-BSR project may be referred.

BSU (BSR Semantic Unit) is concept unambiguously defined, independently of any particular physical representation, and which is semantically complete. It is independent of the process or application in which it is used. It is constructed using BSR semantic components. But BSU is a completely deferent approach from Core Component analyzed in this paper. BSUs represent attributes of Specific Business Entity. In the concept of Common Business Entity, attributes are defined in Common Business Entity and Specific Business Entities inherit the attributes of Common Business Entity.

431

#### 432 **4.4 Naming the component**

433 There are three purposes of naming the component.

- 434 1. To identify the component uniquely in the certain domain if there are no identifiers other than435 the name.
- 436 2. To be recognized easily by human.
- 437 3. To specify the domain structure.

The data element conforming ISO11179 shall have the unique identifier other than the name, therefore, the name of the data element is the primary means of identification of objects and concepts for humans. Otherwise, the name is the only identification for the component in UML. Also the element name is the only identification for the component in XML. However, an element name in XML may have different definitions depending upon the context of the element within the XML structure.

- We need some identification methodology for naming core components even when we start to analyze Business Entities. At the analysis phase, the identifiers should be recognized easily by both humans and computers.
- 447
- 448

## 449 4.5 Segmentation of Core Components

This consideration looks at how it might be possible to use some of the less commonly used featuresof UML to create sets of core components that can be reused in multiple contexts.

- 452 In the paper on *Transformation from EDIFACT to XML* Pharos group members from the EDIFACT
- Transport group suggest that multiple associations should be used to identify the different "roles" aparticular set of data elements play. They state that:
- 455 "A role name defines a task or duty of a class in an association with another class." The example

they use is the qualifier of the Party segment that indicates whether the party concerned is the Buyer, Sollar Consigner Dispatch Party etc.

- 457 Seller, Consignor, Dispatch Party, etc.
- 458 The Pharos document also contains a number of "rules" for the creation of UML models to represent

459 business messages. Rule 5 states:

#### "Pharos Rule 5: Create multiple associations

463Multiple associations are created for message classes that include subsections of the role list<br/>category.

465 466

460 461

462

400

467 468 469

478

479

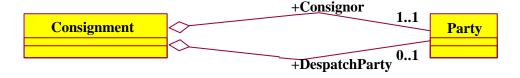


Fig. 8 Example of multiple associations.

- 470
  471 Each of the multiple associations is given a multiplicity and a role name. The role names are
  472 found as data term names in subsections of the role list category.
  473 The role name is used in the generated XML/DTD or XML Schema as a 'group' name for
  474 the attributes in the child message class structure. The UML notation uses a '+' before the
  475 role name to indicate that the role name is 'public', but this symbol will be stripped in the
  476 generated XML/DTD or XML Schema.
  477 The example multiple associations in the figure can be read as:
  - A Consignment has minimum 1 and maximum 1 Consignor
  - A Consignment has minimum 0 and maximum 1 DispatchParty"
- 480 481 Whilst the use of associations to distinguish between the roles played by multiple occurrences of a 482 class that can occur more than once within a document is highly commendable, it cannot be agreed 483 upon with the group name assigned within the context. It is difficult to determine programmatically 484 the context for a class within the XML DTD or Schema. A class with multiple roles should be 485 represented by an abstract class in the UML model, with specific instances of the class being 486 identified in the XML DTD or Schema by means of the name that associates the abstract class with
- the message.
- 488 The section 2.2 (1) takes a more traditional EDI approach to the purpose of these "qualifiers" of data489 element groups. It states:
- 490 The idea that Party is a super class suggested here might be better mapped to the concept of an491 abstract class, because there is no real sense of inheritance needed in this example
- The comment on this concept was that "Assigning a 'role' qualifier to a party specification means 492 493 that the information therein cannot be easily reused as part of an alternative role. If the role of the 494 party is defined by its context, i.e. by its parentage/container within an XML tree, then reusability of 495 Party-related information will be easier to manage. Unfortunately XML trees do not work in the 496 same manner as traditional OO classes. We need to be able to identify things like 497 Order\Buyer\PartyName and Order\Seller\PartyName rather than Order\Parties\Buyer and 498 Order\Parties\Seller, which is what is implied if the party role defines a sub-class of Party. (Note the 499 fact that the XML container needs to be Parties, not Party, if you use a class-based model, as the 500 container is intended to contain information about all parties in current context.)"
- 501 In trying to reconcile the views, the sequence of containment can be expressed as:

502

503	Order

504Parties505Seller506Party Information507Party Identification508....

510 In discussions of the UK Data Harmonization Group at e-centre<sup>uk</sup> it was pointed out that the Parties 511 and Party Information component of this sequence were really just containers that allowed the 512 correct management of data that forms part of the same abstract class. The other three components of 513 the "tree" represent the Business Process being undertaken (Order), the association of the abstract 514 class containing party information with the business process (Seller), and the core components that 515 identify the information to be interchanged for the completion of that part of the business process 516 (Party Identification ...).

The 'real abstract class' in this example is the Party Information abstract class. This may need to take a number of different forms within different messages. For example, in the EB-Simpl model most of the occurrences of data elements making up the Party Information are pre-exchanged, in order that messages used by business processes only need to pass a key of the pre-exchanged information (the Party Identification code). In other messages a wider range of the fields from the Party Information set needs to be interchanged.

523 524

#### 5. Describe the Core Component

Each business entity pattern can be described as the class diagram with the definitions of the
 involved entities. But highly reusable representation classes shall be defined independent from any
 business entities using them.

528

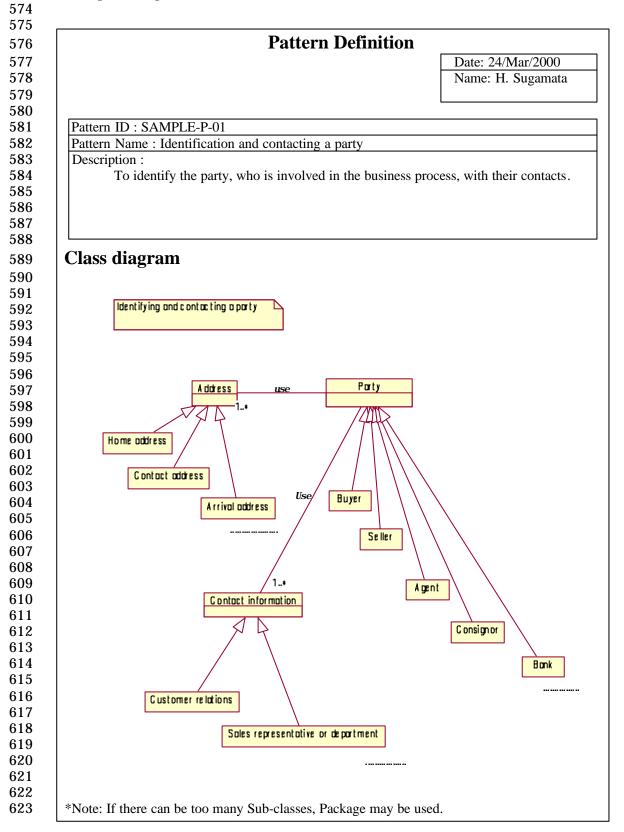
529	(1)	Describing	the	business	entity	pattern
-----	-----	------------	-----	----------	--------	---------

530 The set of the documents describing the business entity pattern is as follows.

531	- Entity pattern definition (One cover sheet)
532	Entity patter ID
533	Entity pattern name
534	Entity pattern description
535	Class diagram representing the entity pattern
536	- Entity definitions (One sheet for each related entity)
537	Entity ID
538	Entity name
539	Entity description
540	Attribute list
541	Attribute ID
542	Attribute name
543	Attribute type (Designating entity or Using representation)
544	Reference identification for designating entity or using representation
545	Parameters for using representation class
546	Attribute description
547	Sub-class list
548	Sub-class ID
549	Sub-class name
550	Sub-class function
551	
552	(2) Describing representation class
553	Each representation class can be described in one sheet.
554	- Representation class definition
555	Representation class ID
556	Representation class name
557	Representation class description
558	Data type definitions
559	Concept (lexical meanings)
560	Concept ID
561	Concept specification
562	Unit of Measure if needed

563	Format (syntactical expressions)
564	Format ID
565	Format specification
566	Character set if needed
567	Value domain list
568	Value domain ID
569	Value domain name
570	Value specification in the domain
571	
572	

### 573 6. Sample description



		<b>Business Er</b>	tity Definition
			Date: 24/Mar/2000
			Name: H. Sugamata
Entity ID: SAMPLE-E	01		
Entity Name: Party	2-01		
Description:			
Unique framework of	authori	ty designating to	act toward some purpose in the business.
Attribute List			
	Туре	Ref.ID	Description
Party identifier	R	SAMPLE-R-01	Identification of the party.
(UN: 7402) Address	Е	(C1, F1, Vn) SAMPLE-E-02	Address of the party.
Audiess	Ľ	SAMIFLE-E-02	Address of the party.
Contact information	Е	SAMPLE-E-03	To identify a person or a department of t
			party, to whom communication should b directed.
*Note1: Type is R (usi	ng Ren	resentation class	or E (designating Business Entity).
			epresentation class in Ref.ID field.
		eter of the Ref.ID	
		eter of the Ref.ID	
Vn :	Parame	eter of the Ref.ID	for Value domain
Sub-Classes			
Sub-Class Name	F	unction	
Buyer			erchandise and/or service is sold.
L	F	Party selling merc	handise to a buyer.
Seller			
Seller etc			
		The sub-classes o	f Party are defined in the code list of the d
			f Party are defined in the code list of the daty function code qualifier) in UN/EDIFAC
	e		f Party are defined in the code list of the d ty function code qualifier) in UN/EDIFAC

		<b>Business En</b>	tity Definition	n
			·	Date: 24/Mar/200
				Name: H. Sugar
Entity ID: SAMPLE-E-	02			
Entity Name: Address				
Description:				
To specify an address.				
Attribute List	•			
Attribute name		Ref.ID	Description	
Address type code	R	SAMPLE-R-02	Code specifying t	he type of an addre
(UN:3131)		(C1, F1, V1)		1 4 4 6 11
Address status code	R	SAMPLE-R-02	Code specifying t	he status of an addr
(UN:3475) Address component	R	(C2, F1, V2) SAMPLE-R-03	Free form description	tion of the compon
description (UN:3286)	K	(C1, Fn, Vn)	of an address.	non of the compon
City name	R	SAMPLE-R-04		
(UN:3164)		(C1, F1, V1)	reality of a city.	
Postal identification	R	SAMPLE-R-02	Code specifying t	he postal zone or ac
Code (UN:3251)		(C3, F2, V3)	see speenjing (	r
Country name code N:3207) (C <sup>2</sup>	R , F3, `	SAMPLE-R-02 V4)	Identification of t	he name of the cou
Country sub-entity	R	SAMPLE-R-04	Name of a country	sub-entity.
name (UN:3228)		(C2, F1, V2)		
Location name code	R	SAMPLE-R-02		he name of the loca
(UN:3225)		(C5, F4, V5)	which is defined	
*Note1: Type is R (usin				
*Note2: 3 Parameters sh				in Ref.ID field.
		eter of the Ref.ID eter of the Ref.ID		
			for Value domain	
VII. I	aranne	Let of the Ref.ID		
Sub-Classes				
Sub-Class Name	F	unction		
Home address	Г	The address is the	home address.	
Contact address	A	Address where co	ntact may be made	
Arrival address	A	Address of arrival		
etc				ed in the code list o
			dress purpose code	
	0	directory. There a	re 7 functions for A	Address in D.00A.

	D	usiness Enti	ty Definition
			Date: 24/Mar/2000
			Name: H. Sugamata
Entity ID: CAMPLE E 02			
Entity ID: SAMPLE-E-03 Entity Name: Contact info		on	
Description:	matr	011	
To identify how to contact directed.	a per	son or a departm	ent to whom communication should be
Attribute List			
Attribute name	Туре		Description
Communication address (UN:3148)	R	(C1, F1, Vn)	A communication address of a departme or a person to whom communication sho be directed.
Department or employee name code (UN:3413)	R	SAMPLE-R-02 (C6, F2, V6)	Code specifying the name of a departme or employee.
Department or employee name (UN:3412)	R	SAMPLE-R-04 (C3, F1, V3)	Name of a department or employee .
			or E (designating Business Entity).
*Note2: 3 Parameters shal Cn : Para Fn : Para	1 be s ameter ameter		presentation class in Ref.ID field. or Concept or Format
*Note2: 3 Parameters shal Cn : Para Fn : Para Vn : Para Sub-Classes	l be s ameter ameter ameter	pecified with Rep r of the Ref.ID for r of the Ref.ID for r of the Ref.ID for	presentation class in Ref.ID field. or Concept or Format
*Note2: 3 Parameters shal Cn : Para Fn : Para Vn : Para	l be s ameter ameter amete	pecified with Rep r of the Ref.ID for r of the Ref.ID for r of the Ref.ID for	presentation class in Ref.ID field. or Concept or Format
*Note2: 3 Parameters shal Cn : Para Fn : Para Vn : Para <b>Sub-Classes</b>	l be s ameter ameter ameter Fun Ind	pecified with Rep r of the Ref.ID for r of the Ref.ID for r of the Ref.ID for ction	presentation class in Ref.ID field. or Concept or Format or Value domain
*Note2: 3 Parameters shal Cn : Para Fn : Para Vn : Para <b>Sub-Classes</b> Sub-Class Name Customer relations Sales representative or	I be sameter ameter ameter Fun Ind The an	pecified with Rep r of the Ref.ID for r of the Ref.ID for r of the Ref.ID for <u>ction</u> ividual responsib e sales representa organization.	presentation class in Ref.ID field. or Concept or Format or Value domain le for customer relations. tive or department contact within
*Note2: 3 Parameters shal Cn : Para Fn : Para Vn : Para <b>Sub-Classes</b> Sub-Class Name Customer relations Sales representative or department	I be sameter ameter ameter ameter Ind Ind The an The eler dire	pecified with Rep r of the Ref.ID for r of the Ref.ID for r of the Ref.ID for ction ividual responsib e sales representa organization. e sub-classes of C ment 3139 (Cont ectory.	presentation class in Ref.ID field. or Concept or Format or Value domain ele for customer relations.

	Representation Class De	Date: 29/Mar/200
		Name:
		Hisanao Sugamata
Representation ID :SAM		
Representation Name : I	dentifier	
	sed to identify and distinguish lue within an identification sc	
Data type		
Concept (lexical meanings	3)	
Specification		Unit of Measure
1 Party identifier		
2		
3		
Format (syntactical expres	ssions)	Character set
1 An35		
2		
3 * The formats of identified	ers are defined in each schema.	
Value domain list		
Domain name	Specification	
1 Duns	Dun & Bradstreet Corpora	ation assigned identifier
2 S.W.I.F.T.	S.W.I.F.T. assigned identif	ier
3etc		
*The identifier schemas a element 1131/3055 in U	are defined in the code list of data N/EDIFACT directory.	

		Date: 29/Mar/200
		Name:
		Hisanao Sugamata
Representation ID :SAMPLE-R-	02	
Representation Name : Code		
Description		
A character string that represent	a member of a set of values.	
Data type		
<b>C</b>		
Concept (lexical meanings) Specification		Unit of Measure
1 Address type code		
2 Address status code		
3 Postal identification code		
4 Country name code		
5 Location name code		
6 Department or employee name	code	
7 Etc		
Format (syntactical expressions)		
Specification		Character set
1 an3		
2 an17		
<u>3 a2</u>		
4 an25		
4 an25		
Value domain list		
Value domain list Domain name	Specification	
Value domain list         Domain name         1 The code list for types of	Specification Refer DE3131 of UN/EDIF	ACT
Value domain list Domain name 1 The code list for types of Address.	Refer DE3131 of UN/EDIF	
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of		
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of         address	Refer DE3131 of UN/EDIF Refer DE3475 of UN/EDIF	ACT
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of         address         3 The code list for postal	Refer DE3131 of UN/EDIF	ACT
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of         address         3 The code list for postal         identification	Refer DE3131 of UN/EDIF Refer DE3475 of UN/EDIF The code lists are assigned	ACT by the country authority
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of         address         3 The code list for postal         identification         4 The code list for countries	Refer DE3131 of UN/EDIF Refer DE3475 of UN/EDIF	ACT by the country authority
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of         address         3 The code list for postal         identification         4 The code list for countries         name	Refer DE3131 of UN/EDIF Refer DE3475 of UN/EDIF The code lists are assigned Codes specified in ISO316	ACT by the country authority 6
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of         address         3 The code list for postal         identification         4 The code list for countries	Refer DE3131 of UN/EDIF Refer DE3475 of UN/EDIF The code lists are assigned	ACT by the country authority 6
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of         address         3 The code list for postal         identification         4 The code list for countries         name         5 The code list for locations	Refer DE3131 of UN/EDIF Refer DE3475 of UN/EDIF The code lists are assigned Codes specified in ISO316 UNLOCODE specified UN	ACT by the country authority 5 I/ECE recommendation 1
Value domain list         Domain name         1 The code list for types of         Address.         2 The code list for status of         address         3 The code list for postal         identification         4 The code list for countries         name	Refer DE3131 of UN/EDIF Refer DE3475 of UN/EDIF The code lists are assigned Codes specified in ISO316	ACT by the country authority 5 I/ECE recommendation 1

ľ	Representation Class Definiti	
	_	Date: 29/Mar/200 Name:
		Hisanao Sugamata
	L	Thsanao Sugamat
Representation ID :SAMPI		
	dress component description	
Description		
A description of the compo	onent of an address.	
Data type		
Concept (lexical meanings)		
Specification		Unit of Measure
1 Address component descr	ription	
2		
Format (syntactical expressi	ons)	
Specification		Character set
1 Address format : Street n		
	, road type, road name in this sequence	
	pe, road name, number in this sequence	
4 Address format: Post offi	ice box	
5 Address format: Unstruct	ured address	
5 Address format: Unstruct		
	ame followed by number, building, suite	
	ame followed by number, building, suite	
6 Address format: Street na	ame followed by number, building, suite ute number	
6 Address format: Street na 7 Address format: Rural ro	ame followed by number, building, suite ute number ice drawer number	
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi	ame followed by number, building, suite ute number ice drawer number	
<ul><li>6 Address format: Street na</li><li>7 Address format: Rural ro</li><li>8 Address format: Post offi</li><li>9 Address format: Building</li></ul>	ame followed by number, building, suite ute number ace drawer number g name followed by suite	
<ul> <li>6 Address format: Street na</li> <li>7 Address format: Rural ro</li> <li>8 Address format: Post offi</li> <li>9 Address format: Building</li> <li>Note:1 Address formats ar</li> </ul>	ame followed by number, building, suite ute number the drawer number g name followed by suite re specified in the code list of the data	
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF	ame followed by number, building, suite ute number the drawer number g name followed by suite re specified in the code list of the data	
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF	ame followed by number, building, suite ute number are drawer number g name followed by suite re specified in the code list of the data ACT	
<ul> <li>6 Address format: Street na 7 Address format: Rural ro</li> <li>8 Address format: Post offi</li> <li>9 Address format: Building</li> <li>Note: 1 Address formats ar element 3477 in UN/EDIF</li> </ul>	ame followed by number, building, suite ute number ce drawer number g name followed by suite re specified in the code list of the data ACT	
<ul> <li>6 Address format: Street na 7 Address format: Rural ro</li> <li>8 Address format: Post offi</li> <li>9 Address format: Building</li> <li>Note: 1 Address formats ar element 3477 in UN/EDIF</li> </ul> Value domain list Domain name 1 Address component	ame followed by number, building, suite ute number are drawer number g name followed by suite re specified in the code list of the data ACT	
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1	ame followed by number, building, suite ute number ice drawer number name followed by suite re specified in the code list of the data ACT Specification Street name followed by number	
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1 2 Address component	ame followed by number, building, suite ute number ce drawer number g name followed by suite re specified in the code list of the data ACT	
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1 2 Address component Description -2	ame followed by number, building, suite ute number ce drawer number g name followed by suite re specified in the code list of the data ACT Specification Street name followed by number Number, road type, road name in	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component	ame followed by number, building, suite ute number ice drawer number name followed by suite re specified in the code list of the data ACT Specification Street name followed by number	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3	ame followed by number, building, suite ute number ce drawer number g name followed by suite re specified in the code list of the data ACT Specification Street name followed by number Number, road type, road name in Road type, road name, number in	n this sequence
<ul> <li>6 Address format: Street na 7 Address format: Rural ro</li> <li>8 Address format: Post offi</li> <li>9 Address format: Building</li> <li>Note: 1 Address formats ar element 3477 in UN/EDIF.</li> </ul> Value domain list Domain name <ol> <li>1 Address component</li> <li>Description -1</li> <li>2 Address component</li> <li>Description -2</li> <li>3 Address component</li> <li>Description -3</li> <li>4 Address component</li> </ol>	ame followed by number, building, suite ute number ce drawer number g name followed by suite re specified in the code list of the data ACT Specification Street name followed by number Number, road type, road name in	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF 7 7 7 8 7 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component	ame followed by number, building, suite ute number ce drawer number g name followed by suite re specified in the code list of the data ACT Specification Street name followed by number Number, road type, road name in Road type, road name, number in	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component Description -5	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box         Post office box	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component Description -5 6 Address component	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF /alue domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component Description -5 6 Address component Description -6	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box         Post office box         Street name followed by number	n this sequence
<ul> <li>6 Address format: Street na 7 Address format: Rural ro</li> <li>8 Address format: Post offi</li> <li>9 Address format: Building</li> <li>Note:1 Address formats ar element 3477 in UN/EDIF.</li> </ul> Value domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component Description -5 6 Address component Description -6 7 Address component	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box         Post office box	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF /alue domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component Description -5 6 Address component Description -6	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box         Post office box         Street name followed by number	n this sequence
<ul> <li>6 Address format: Street na 7 Address format: Rural ro</li> <li>8 Address format: Post offi</li> <li>9 Address format: Building</li> <li>Note:1 Address formats ar element 3477 in UN/EDIF.</li> </ul> Value domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component Description -5 6 Address component Description -6 7 Address component	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box         Post office box         Street name followed by number	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note: 1 Address formats ar element 3477 in UN/EDIF. /alue domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component Description -5 6 Address component Description -6 7 Address component Description -7	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box         Street name followed by number         ACT	n this sequence
6 Address format: Street na 7 Address format: Rural ro 8 Address format: Post offi 9 Address format: Building Note:1 Address formats ar element 3477 in UN/EDIF Value domain list Domain name 1 Address component Description -1 2 Address component Description -2 3 Address component Description -3 4 Address component Description -4 5 Address component Description -5 6 Address component Description -6 7 Address component Description -7 8 Address component	ame followed by number, building, suite         ute number         ice drawer number         g name followed by suite         re specified in the code list of the data         ACT         Specification         Street name followed by number         Number, road type, road name in         Road type, road name, number in         Post office box         Street name followed by number         ACT	n this sequence n this sequence , building, suite

-	esentation Class Definit	Date: 29/Mar/200
	Ē	Name:
		Hisanao Sugamat
Representation ID :SAMPLE-R-0	)4	
Representation Name : Name		
	titutes the distinctive desigr ept. What the person, object, ed.	
Data type		
Concept (lexical meanings)		
Specification		Unit of Measure
1 City name		
2 Country sub-entity name		
3 Department or employee name		
Specification 1 an35 2 3		Character set
Value domain list Domain name	Specification	
1 City name	Name of a city in alphabetic cha	iracters
2 Country sub-entity name	Country sub-entity name in alph	abetic characters
3 Department or employee name	Name of a department or emplo	yee in alphabetic
		he relevant code s

		presentation Class Defin	Date: 29/Mar/200
			Name:
			Hisanao Sugamata
	Representation ID :SAMPLE-	-R-05	
	Representation Name : Comm		
	Description:		
	A communication contact add should be directed.	ress of a department or a person	to whom communicatio
	Data type Concept (lexical meanings)		
-	Specification		Unit of Measure
	1 Communication address		
	2		
	ormat (syntactical expressions	.)	
	Specification	/	Character set
	1 an512		
	2		
	3		
	Value domain list		
	Domain name	Specification	
	1 International telephone	Telephone number, including as required, for voice or data beyond the border of a count	transmission by teleph
	2 World Wide Web	Data exchange via the World	
	3 Electronic mail	Exchange of mail by electron	nic means.
		efined in the data element 3155 ir	

## 1046 7. Instructions for capturing ebXML Core Component definitions

1047

47

1048 The following instructions describe how Microsoft's Internet Explorer 5.0 (IE5) can be
1049 used to capture core component definitions as XML files.
1050

1051 WARNING: This process will not work on Netscape Explorer or on older versions of
1052 Internet Explorer as it relies on Microsoft specific extensions to HTML.
1053

1054 The UML metamodel for ebXML core components recognizes the following types of 1055 record:

1056

1060

- Patterns: UML models that define a set of related data entities, and the associations between them
  Entities: UML model components that identify sets of related components.
  - Entities: UML model components that identify sets of related components, including data elements with a specific data representation
- 1061- Date Representations: the set of data patterns or code lists that can be1062used to capture a particular component of an entity
- 1063- Code Sets: A list of permitted values that can be used to complete a<br/>component, together with descriptions of their meaning
- 1065- Data Formats: Details of constraints to be placed on the contents1066particular components of a pattern.
- 1067

1068 The HTML forms provided in this suite allow each of these data types to be recorded. 1069 To allow the relationships between forms to be clearly identified the basic UML model 1070 has been extended by adding elements that reference the ID of the next lowest level of 1071 component in the model. (This means that once a component, data representation, code 1072 list or data format has been defined once it need only be referenced in subsequent 1073 models.)

1074

For each of the above record types there is an form with the appropriate name
(pattern.htm, entity.htm, representation.htm, CodeSet.htm and DataFormat.htm).
Each form contains buttons that allow the current contents to be submitted for storage,
reset so that a new entry of the same type can be made or request a form for a record at
a lower level in the metamodel.

1080

1081 The package has been designed to be run within a directory called ebxml within the My 1082 Documents section of your C drive. If wish to use another drive or root directory you 1083 should search each of the above files for any occurrences of the string "C:/My 1084 Documents/ebxml/" and replace this with the appropriate identifier for the directory 1085 you wish to use.

1086

1087 Within the directory that is used to contain the forms you will need to create five 1088 subdirectories prior to using the forms. These subdirectories should be labelled 1089 patterns, entities, representations, codesets and dataformats respectively. A further 1090 directory, called class, can also be defined to store class diagrams, etc. (Directories with 1091 these names may or may not have been created for you when you unzipped the source 1092 files, depending on your settings of your file unzipper.)

1093

Because the forms submit their contents to the local file store rather than to a remote
directory you will need to ensure that IE5 has been set up to permit this. In the Tools
Menu select the Internet Options entry and then the Security tab. If your security level
is High you will not be able to use these forms. If it is set to Medium you will be asked

to confirm that each record may be written to disc. If it is Low you will be able to write
the files without having to confirm each one, but must take care if using the Internet. (I
find that the Medium level, which requires me to confirm before writing but still
provides for a safe level of Internet access is an adequate compromise.)

## 1103 **7.1 Defining a Pattern**

1104

1102

1105 The following figure shows the fields used to define a new pattern for use as a core 1106 component. The status line at the foot of the form will change as you move from field to 1107 field to provide you with information about the type of data to be entered in each field.

ebXML Data Representation Class - Microsoft Internet Explorer	_ 🗆 ×
j <u>F</u> ile <u>E</u> dit <u>V</u> iew F <u>a</u> vorites <u>I</u> ools <u>H</u> elp	-
j ← Back + → - 🙆 🖄 🖓 + 🎒 💽 + 🗐 🧭 History 📷 Favorites 📿 Search	
Address 🚱 C:\My Documents\ebxml\pAttern.htm	ểGo
ebXML Pattern Definition	<b>^</b>
Date of Submission: 2000-05-31	-
Submitting Organization: UK-DHG	
Contact E-mail Address: mtbryan@diffuse.org	
Industry Sector: Retailing Business Process Purchasing Pattern ID: Party-001 Directory for Pattern storage: C:/My Documents/ebxml/patterns/ Pattern Name: Party Pattern Description: Party involved in Simpl-eb purchase	
URL of Class Diagram: C:/My Documents/ebxml/class/Party-001.gif URL of XMI Representation of model: C:/My Documents/ebxml/class/Party-001.xmi	
ID to be assigned to root Component of Pattern: Party-001	-
🛃 Assign Unique ID which can be used as filename of pattern	//

1108

1110

1111

1112

1113 1114

1109 The fields provided are:

- Date of Submission: enter an ISO 8601 conformant date using the format CCYY-MM-DD here
- Submitting Organization: enter the name of the submitting organization here Contact E-mail Address: enter e-mail address to which questions can be submitted
- 1115- Industry Sector: identify which communities the pattern is expected to be1116used by (using SIC codes where available)
- 1117- Business Process: record business process(es) to which pattern applies1118(using SIC codes where available)
- 1119-Pattern ID: Enter a unique identifier for the pattern. (This will form the1120file name of the pattern definition.)
- 1121-Directory for pattern storage: Should indicate the path required to reach1122the required directory. (When the default directory set up is being used1123this entry should not need to be updated from the pre-assigned values.)

- 1124-Pattern Name: Name to be used to identify the pattern. (May or may not1125be the same as the Pattern ID.)
- 1126-Pattern Description: Enter description of role of pattern that allows it to be1127distinguished from other patterns.
- 1128-URL of Class Diagram: Enter URL of file containing printable version of1129class diagram. (For ease of use on the web we recommend this be a GIF1130file. If this is stored in the class subdirectory within you ebxml directory1131then all you need to do is to replace the ??? in the default name displayed1132with the filename.)
- 1133-URL of XMI Representation of model: If an XMI representation of the1134model is available for interchange enter the appropriate file reference1135here: otherwise delete the default value.
- 1136- ID to be assigned to root Component of Pattern: Indicate which ID you1137expect to assign to the Entity which will form the root of the pattern
- 1138- Directory that will be used for Entity storage: Should indicate the path1139required to reach the required directory. (When the default directory set1140up is being used this entry should not need to be updated from the pre-1141assigned values.)
- 1142

1143 When all relevant fields have been completed click on the Submit this Pattern button.

1144 This will cause the inbuilt program to store an XML record pattern and then display

1145 the contents of the file as shown below.

🚰 C:\My Documents\ebxml\patterns\Party-001.xml - Microsoft Internet Explorer 📃 🔲
j <u>F</u> ile <u>E</u> dit <u>V</u> iew F <u>a</u> vorites <u>I</u> ools <u>H</u> elp
📙 🕁 Back 🔹 🤿 🖉 🚳 🛃 🛃 📰 🚍 🧭 🍏 History 💿 Favorites 🛛 Search
Address 🖭 C:\My Documents\ebxml\patterns\Party-001.xml
<pre><?xml version="1.0" ?> <!DOCTYPE CoreComponents (View Source for full doctype)>     - <corecomponents> - <pattern></pattern></corecomponents></pre>
🔊 Done 📃 My Computer

#### 1146

1147 Once you have ascertained that this file correctly records the details of your pattern the1148 window can be dismissed from the screen.

1149

The Pattern form ends with a button that allows you to "Create Entity Definition for
Root Component". Clicking on this button will call up the form needed to record each of
the entities defined within the pattern.

1153

## 1154 **7.2 Defining an Entity**

The following figures show the fields used to define a new entity for use within a core component. The status line at the foot of the form will change as you move from field to field to provide you with information about the type of data to be entered in each field.

🖉 ebXML Business Entity Definition -	- Microsoft Internet Explorer	
<u>File Edit View Favorites Tools</u>		
<u> </u>	→ 🔄 🖸 + 🧮 🍏 History 👔 Favorites 🔞 Search	Links »
Address C:\My Documents\ebxml\ent		▼ 🖉 Go
	NY COMP	
AVM	II Dusings Entity Definition	- 1
edawi	IL Business Entity Definition	
	Date of Submission:	
	Submitting Organization:	
	Contact E-mail Address:	
Industry Sector:	Business Process	
Entity ID:	Directory for Entity storage: C:/My Documents/ebxml/entities/	
	Directory for Entity storage, jetwy Documents/ebonitendies/	
Entity Name: Entity Description:		
	*	
	_	
	¥.	
1		- L P
Done	🛄 My Comp	outer

ebXML Business Entity D	efinition - Microsoft Intern	et Explorer			
<u>F</u> ile <u>E</u> dit <u>V</u> iew F <u>a</u> vorite	s <u>T</u> ools <u>H</u> elp				
🛛 🖛 Back 🔻 🔿 👻 🛃	십 B- ∌ R • E	💮 History 🛛 🙀 Fav	orites 🧿	Search	Links »
🛛 Address 餐 C:\My Documents	\ebxml\entity.htm				▼ 🖉 Go
Embedded Components					<b>_</b>
Name	Туре	Identifier <sup>1</sup>	Card. <sup>2</sup>	Description	
	Entity Definition	]			
	Entity Definition				
	Entity Definition				
	Entity Definition				
	Entity Definition				
	Entity Definition				
	Entity Definition				
	Entity Definition				
	Entity Definition				
	Entity Definition				
	Entity Definition				
	Entity Definition				<b>_</b>
🥙 Done					🛄 My Computer 🛛 🎼

a ebXML Business Entity [	Definition - Microsoft Internet Explorer	
		-
← Back → → ∞ 🙆	] 🖓 🗟 🕶 🛃 🗸 🗐 🧭 History 💿 Favorites 📿 Search 🛛 🗍 Lin	ks »
Address 🛃 C:\My Document	ts\ebxml\entity.htm 🔽 🔗	'Go
<u>, , , , , , , , , , , , , , , , , , , </u>		
Known Examples of Use of I	Entity	
Application Name	Application Description Constraints on use of Entity <sup>3</sup>	
		-
1		•
🗳 Done	My Computer	_/

- The fields provided in the first part of the form are: Date of Submission: enter an ISO 8601 conformant date using the format CCYY-MM-DD here
- Submitting Organization: enter the name of the submitting organization -here

1168	- Contact E-mail Address: enter e-mail address to which questions can be
1169	submitted
1170	- Industry Sector: identify which communities the pattern is expected to be
1171	used by (using SIC codes where available)
1172	- Business Process: record business process(es) to which pattern applies
1173	(using SIC codes where available)
1174	- Entity ID: Enter a unique identifier for the entity. (This will form the file
1175	name of the entity definition.)
1176	- Directory for entity storage: Should indicate the path required to reach the
1177	required directory. (When the default directory set up is being used this
1178	entry should not need to be updated from the pre-assigned values.)
1179	- Entity Name: Name to be used to identify the entity. (May or may not be
1180	the same as the Entity ID.)
1181	- Entity Description: Enter description of role of entity that allows it to be
1182	distinguished from other core components.
1183	
1184	Each line in the section headed Embedded Components has the following fields:
1185	- Name: Name of component in form suitable for use in XML DTD (i.e.
1186	starting with a letter and containing no spaces)
1187	- Type: Type of component (either Entity Definition if there are embedded
1188	components or Data Representation if this component is a root one
1189	designed to transfer data between systems)
1190	- Identifier: Indicate which ID you expect to assign to the
1191	Entity/Representation when you define it (this will create a cross reference
1192	to the definition you will create at a subsequent stage using either another
1193	copy of this form or the form for recording data representations.)
1194	- Card. If the embedded component is optional and/or repeatable the
1195	cardinality of this component with respect to its parent (i.e. how many
1196	times it can occur within the parent) as defined in the UML model should
1197	be recorded here, expressed as $01$ (optional), $0*$ (optional and
1198	repeatable), 1* (required and repeatable) or mn (m=min, n=max no of
1199	occurrences).
1200	- Description: Brief description of role of component within Entity.
1201	
1202	If the entity is a pattern that is intended to be sub-classed or used as an abstract class
1203	that is associated with concrete classes, examples of the intended use of the entity can
1204	be recorded in the section headed Known Examples of Use of Entity. The following
1205	fields can be used for each example:
1206	- Application Name: Name by which sub-class or association is known
1207	- Application Description: Description of purpose of application
1208	- Constraints on use of Entity: Details of any constraints that apply to the
1209	use of the entity within this application (e.g. components that must or may
1210	not be used for this application of the pattern/entity.)
1211	
1212	Note: At present there is no formal language for defining such constraints but in
1213	future it is anticipated that text based descriptions entered initially will be replaced by
1214	machine processable XML descriptions of the required constraints at a later date.
1215	
1216	The buttons at the foot of the Entity Definition form are:
1217	- Submit this Entity:
1218	- New Entity Definition: Resets the form so that details of another
1219	component can be defined. (Each embedded component defined using

1220 Entity Description in the Type field requires completion of a separate 1221 Entity Description form.)

- Create Data Representation Definition: Calls up the form needed to record a Data Representation. (Each embedded component defined using Data Representation in the Type field requires completion of a Data Representation form if the identified format has not previously been defined.)
- 1227

1229

## 1228 **7.3 Defining a Data Representation**

1230 A data representation identifies one or more data formats and/or code sets that can be
1231 used to record a particular type of data stored within a core component.
1232

Note: While normally a data representation will only define data of the same type (e.g. one or more data formats or one or more code lists) there are cases where both a code list and a data format will be required (e.g. to define a list of known codes plus a pattern that can be used to extend the list where appropriate.)

1237

1238 The following figure shows the fields used to define a new data representation for use 1239 within a core component. The status line at the foot of the form will change as you 1240 move from field to field to provide you with information about the type of data to be 1241 entered in each field.

🗿 ebXML Data Representation Class - Microsoft Internet Explorer 📃	
<u>Eile E</u> dit <u>V</u> iew F <u>a</u> vorites <u>T</u> ools <u>H</u> elp	
🖕 Back 🗸 🤿 🖉 🖓 📴 🚭 🖉 🗲 🧮 🧭 History 🕋 Favorites 🎯 Search 🛛 🗍	nks »
Address 🚳 C:\My Documents\ebxml\Representation.htm	ờGo
ebXML Data Representation Class	1
Date:	
Submitter:	
Data Representation ID: Directory for Representation storage: C:/My Documents/ebxml/representation	
Data Representation Name:	
Data Representation Purpose Description:	
Type Code Set or Data Format ID Code Set  Code Set Code Set	•
🖉 Done 🥢 🛄 My Computer	

- 1243 The fields provided on this form are:
- 1244 Date of Submission: enter an ISO 8601 conformant date using the format

1245	CCYY-MM-DD here
1246	- Submitting Organization: enter the name of the submitting organization
1247	here
1248	- Contact E-mail Address: enter e-mail address to which questions can be
1249	submitted
1250	- Date Representation ID: Enter a unique identifier for the data
1251	representation. (This will form the file name of the data representation
1252	definition.)
1253	- Directory for Representation storage: Should indicate the path required to
1254	reach the required directory. (When the default directory set up is being
1255	used this entry should not need to be updated from the pre-assigned
1256	values.)
1257	- Data Representation Name: Name to be used to identify the data
1258	representation. (May or may not be the same as the Date Representation
1259	ID.)
1260	- Data Representation Description: Enter description of role of data
1261	representation that allows it to be distinguished from other
1262	representations.
1263	- Type: Select Code Set if the representation is to reference a code set, or
1264	Data Format if it is to reference a data format
1265	- Code Set or Data Format ID: Enter the unique identifier to be assigned to
1266	the referenced definition.
1267	
1268	The form ends with the following buttons:
1268 1269	- Submit this Representation: Creates XML record of form and displays this
1268 1269 1270	- Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise
1268 1269 1270 1271	- Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that
1268 1269 1270 1271 1272	- Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)
1268 1269 1270 1271 1272 1273	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another</li> </ul>
1268 1269 1270 1271 1272 1273 1274	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format.</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires completion of a Domain Value Data Format form if the identified format</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires completion of a Domain Value Data Format form if the identified format has not previously been defined.)</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires completion of a Domain Value Data Format form if the identified format</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires completion of a Domain Value Data Format form if the identified format has not previously been defined.)</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires completion of a Domain Value Data Format form if the identified format has not previously been defined.)</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires completion of a Domain Value Data Format form if the identified format has not previously been defined.)</li> </ul> 7.4 Defining a Data Format
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires completion of a Domain Value Data Format form if the identified format has not previously been defined.)</li> <li>The following figure shows the fields used to define a code set for use within a core component representation. The status line at the foot of the form will change as you move from field to field to provide you with information about the type of data to be</li> </ul>
1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287	<ul> <li>Submit this Representation: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return to source and use the Back button to return to your entries so that they can be corrected and resubmitted.)</li> <li>Define another Representation: Resets the form so that details of another representation can be defined.</li> <li>Define Code Set: Calls up the form needed to record a Code Set. (Each format defined using Code Set in the Type field requires completion of a Domain Value Code Set form if the identified format has not previously been defined.)</li> <li>Define Data Format: Calls up the form needed to record a Data Format. (Each format defined using Data Format in the Type field requires completion of a Domain Value Data Format form if the identified format has not previously been defined.)</li> </ul> 7.4 Defining a Data Format

🚰 ebXML Value Domain Data Format Definition - Microsoft Internet Explorer	
File Edit View Favorites Tools Help	
(← Back - → - ② ② ♂ ♂ Ea- ④ N - E ③ History  Favorites ② Search	Links »
Address @ C:\My Documents\ebxml\DataFormat.htm	▼ ∂Go
	·
ebXML Value Domain Data Format Definition	
Date:	
Submitter	
Data Format ID: Directory for Data Format storage: C:/My Documents/ebxml/dataforma	ats/
Concept Name:	
Concept Description:	
<b>_</b>	
Data Format Type: None	_
Code Set Character Set: ISO 10646/Unicode 💌	
Data Format Definition:	
	-
🖉 Done 📃 My Comp	uter 🗸

1293

1294

1297

1298 1299

1300

1292 The fields provided on this form are:

- Date of Submission: enter an ISO 8601 conformant date using the format CCYY-MM-DD here
- 1295-Submitting Organization: enter the name of the submitting organization1296here
  - Contact E-mail Address: enter e-mail address to which questions can be submitted
    - Data Format ID: Enter a unique identifier for the data format. (This will form the file name of the data format definition.)
- 1301-Directory for Data Format Set storage: Should indicate the path required1302to reach the required directory. (When the default directory set up is being1303used this entry should not need to be updated from the pre-assigned1304values.)
- 1305- Data Format Concept Name: Enter name to be used to identify the concept1306- behind code list. (May or may not be the same as the Data Format ID.)
- 1307- Data Format Concept Description: Enter description of role of data format1308that allows it to be distinguished from other data formats.
- 1309- Data Format Type: If code set has pattern based on one of the known data1310format languages select relevant entry from list. Otherwise select None
- 1311 Character Set: If relevant, select controlling character set from list supplied.
- 1313- Data Format Definition: If relevant, enter pattern that defines format of<br/>codes set values using language identified in Code Set Data Format field

1315	
1316	The form ends with the following buttons:
1317	- Submit this Data Format: Creates XML record of form and displays this in
1318	a separate window. (Dismiss window if record is accurate: otherwise
1319	return to source and use the Back button to return to your entries so that
1320	they can be corrected and resubmitted.)
1321	- Define another Data Format: Resets the form so that details of another
1322	code set can be defined.
1323	- Define Code Set: Calls up the form needed to define a Code Set.
1324	- New Entity Definition: Calls up the form needed to define an Entity.
1325	- Create Data Representation Definition Calls up the form needed to define
1326	a Data Representation.
1327	
1328	7.5 Defining a Code Set
1329	
1330	The following figures show the fields used to define a code set for use within a core

1330 The following figures show the fields used to define a code set for use within a core 1331 component representation. The status line at the foot of the form will change as you 1332 move from field to field to provide you with information about the type of data to be 1333 entered in each field.

1334

🖉 ebXML Value Domain Code Set Definition - Microsoft Internet Explorer	. 🗆 ×
∫ <u>F</u> ile <u>E</u> dit <u>V</u> iew F <u>a</u> vorites <u>T</u> ools <u>H</u> elp	
← Back → → → 🙆 🛐 🖓 🛃 → 🎒 💽 → 🗐 🧐 History 🕋 Favorites 📿 Search	Links »
Address 🖉 C:\My Documents\ebxml\CodeSet.htm	∂Go
ebXML Value Domain Code Set Definition	-
Date of Submission:	
Submitting Organization:	
Contact E-mail Address:	
	-
Code Set ID: Directory for Code Set storage: C:/My Documents/ebxml/codesets/	
Control Agency: Agency Assigned Name:	
Code Set Concept Name:	
Code Set Concept Description:	
_	
Code Set Data Format Type: None	
Code Set Character Set: ISO 10646/Unicode 💌	
Code Set Data Format Definition:	
🖉 Done My Computer	

🖉 ebXML Value Domain Co	le Set Definition - Microsoft Internet Explorer	_ 🗆 ×
∫ <u>F</u> ile <u>E</u> dit <u>V</u> iew F <u>a</u> vorite	s <u>T</u> ools <u>H</u> elp	
] 🖛 Back 🔻 🔿 👻 🙋	🗥 🛛 🛃 🍜 💽 🔹 📃 🛛 🎯 History 🕋 Favorites 🛛 Q Search	] Links »
Address 🖉 C:\My Documents	\ebxml\CodeSet.htm	🝷 🤗 Go
Permitted Value	Description of Meaning of Value	Language 🔺
		EN
1		
🥙 Done	🛄 My Comp	outer //

1338 The fields provided on this form are:

1338	The fields provided on this form are:
1339	- Date of Submission: enter an ISO 8601 conformant date using the format
1340	CCYY-MM-DD here
1341	- Submitting Organization: enter the name of the submitting organization
1342	here
1343	- Contact E-mail Address: enter e-mail address to which questions can be
1344	submitted
1345	- Code Set ID: Enter a unique identifier for the code set. (This will form the
1346	file name of the code set definition.)
1347	- Directory for Code Set storage: Should indicate the path required to reach
1348	the required directory. (When the default directory set up is being used
1349	this entry should not need to be updated from the pre-assigned values.)
1350	- Code Set Control Agency: Enter name of agency responsible for adding
1351	new entries to code set.
1352	- Code Set Concept Name: Enter name to be used to identify the concept
1353	behind code list. (May or may not be the same as the Code Set ID.)
1354	- Code Set Concept Description: Enter description of role of code set that
1355	allows it to be distinguished from other code sets.
1356	- Code Set Data Format Type: If code set has pattern based on one of the
1357	known data format languages select relevant entry from list. Otherwise
1358	select None
1359	- Code Set Character Set: If relevant, select controlling character set from
1360	list supplied.
1361	- Code Set Data Format Definition: If relevant, enter pattern that defines
1362	format of codes set values using language identified in Code Set Data
1363	Format field
1364	- Permitted Value: Enter permitted code set values in the fields in this
1365	column.

- 1366 Description of Meaning of Value: Enter description of how value is to be \_ 1367 interpreted. Language: If description in not in English, enter ISO 639 code for language 1368 \_ used for description. 1369 1370 The form ends with the following buttons: 1371 1372 Submit this Code Set: Creates XML record of form and displays this in a separate window. (Dismiss window if record is accurate: otherwise return 1373 to source and use the Back button to return to your entries so that they 1374 can be corrected and resubmitted.) 1375 Define another Code Set: Resets the form so that details of another code 1376 \_ set can be defined. 1377 1378 Define Data Format: Calls up the form needed to record a Data Format. New Entity Definition: Calls up the form needed to define an Entity. 1379 1380 Create Data Representation Definition Calls up the form needed to define \_ a Data Representation. 1381 1382 1383 **Troubleshooting** The forms described in this document have not been fully tested to date. If you 1384 1385 encounter any problems using them please contact their author, Martin Bryan, via email at mtbryan@sgml.u-net.com (please be patient as I am traveling a lot during 1386
- 1387 May/June so may not be able to respond as fast as you would like.)
- 1388 1389