Organization for the Advancement of Structured Information Systems

Business Transaction Protocol

An OASIS Committee Specification

CURRENT STATUS : internal committee draft

Version 1.0 [0.9.1.2]

DD Mmm 2001 [30 January 2002 10:00]

Working draft 0.1 (pre-London)	14 June 2001
Working draft 0.2 (London)	18 June 2001
Working draft 0.3a (circulated)	12 July 2001
Working draft 0.3c (circulated)	20 July 2001
Working draft 0.4 (circulated; incorporates PRF material)	25 July 2001
Working draft 0.6 (State tables)	31 August 2001
Working Draft 0.9	24 October 2001
Working Draft 0.9.0.1 – minor editorials issues applied	16 November 2001
Working Draft 0.9.0.2 – issue resolutions balloting to 10 Dec 2001	4 December 2001
Working Draft 0.9.0.3 – possible solution to msging issues	11 December 2001
Working Draft 0.9.0.4 – issue 79 solution, revise msging issues	12 January 2002
Working Draft 0.9.1 – includes all issues agreed 16 Jan 2002, and 82 (deferred)	18 January 2002
Working Draft 0.9.1.1 – format changes and proposed soln 77,78, 17.	27 January 2002
Working Draft 0.9.1.2 – xml changes, new schema, and issue 74	30 January 2002

Change marks relative to 0.9.1

- Copyright and related notices 14 15 16 Copyright © The Organization for the Advancement of Structured Information Standards 17 (OASIS), 2001. All Rights Reserved. 18 19 This document and translations of it may be copied and furnished to others, and derivative 20 works that comment on or otherwise explain it or assist in its implementation may be 21 prepared, copied, published and distributed, in whole or in part, without restriction of any 22 kind, provided that the above copyright notice and this paragraph are included on all such 23 copies and derivative works. However, this document itself may not be modified in any way, 24 such as by removing the copyright notice or references to OASIS, except as needed for the 25 purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required 26 27 to translate it into languages other than English. 28 29 The limited permissions granted above are perpetual and will not be revoked by OASIS or its 30 successors or assigns. 31 32 This document and the information contained herein is provided on an "AS IS" basis and 33 OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT 34 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION 35 HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF 36 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. 37 38 39 OASIS takes no position regarding the validity or scope of any intellectual property or other 40 rights that might be claimed to pertain to the implementation or use of the technology 41 described in this document or the extent to which any license under such rights might or 42 might not be available; neither does it represent that it has made any effort to identify any 43 such rights. Information on OASIS's procedures with respect to rights in OASIS 44 specifications can be found at the OASIS website. Copies of claims of rights made available 45 for publication and any assurances of licenses to be made available, or the result of an attempt 46 made to obtain a general license or permission for the use of such proprietary rights by 47 implementors or users of this specification, can be obtained from the OASIS Executive 48 Director. 49 50 OASIS invites any interested party to bring to its attention any copyrights, patents or patent 51 applications, or other proprietary rights which may cover technology that may be required to 52 implement this specification. Please address the information to the OASIS Executive 53 Director.
- 54

54	Acknowledgements
55	
56	Employees of the following companies participated in the finalization of this specification as
57	members of the OASIS Business Transactions Technical Committee:
58	
59	BEA Systems, Inc.
60	Bowstreet, Inc.
61	Choreology Ltd.
62	Entrust, Inc.
63	Hewlett-Packard Co.
64	Interwoven Inc.
65	IONA Technologies PLC
66	SeeBeyond Inc.
67	Sun Microsystems Computer Corp.
68	Talking Blocks Inc.
69	
70	The primary authors and editors of the main body of the specification were:
71	
72	Alex Ceponkus (<u>alex@ceponkus.org</u>)
73	Peter Furniss (<u>peter.furniss@choreology.com</u>)
74	Alastair Green (alastair.green@choreology.com)
75	
76	Additional contributions to its writing were made by
77	
78	Sanjay Dalal (<u>sanjay.dalal@bea.com</u>)
79	Mark Little (<u>mark_little@hp.com</u>)
80	
81	We thank Pal Takacsi-Nagy of BEA Systems Inc for his efforts in chairing the Technical
82	Committee, and Karl Best of OASIS for his guidance on the organization of the Committee's
83	work.
84	
85	
86	
87	In memory of Ed Felt
88	
89	Ed Felt of BEA Systems Inc. was an active and highly valued contributor to the work of the
90	OASIS Business Transactions Technical Committee.
91	
92	His many years of design and implementation experience with the Tuxedo system,
93	Weblogic's Java transactions, and Weblogic Integration's Conversation Management
94	Protocol were brought to bear in his comments on and proposals for this specification.
95	
96	He was killed in the crash of the hijacked United Airlines flight 93 near to Pittsburgh,
97	on 11 September 2001.
98	
99	

99	Typographical and Linguistic Conventions and Style
100	
101	The initial letters of words in terms which are defined (at least in their substantive or
102	infinitive form) in the Glossary are capitalized whenever the term used with that exact
103	meaning, thus:
104	6,
105	Cancel
106	Participant
107	Application Message
108	rippiloution message
109	The first occurrence of a word defined in the Glossary is given in bold, thus:
110	The first occurrence of a word defined in the Glossary is given in bold, thus.
111	Coordinator
112	Coordinator
112	Such words may be given in bold in other contexts (for example, in section headings or
113	captions) to emphasize their status as formally defined terms.
114	capitons) to emphasize their status as formany defined terms.
115	The names of electronic PTD protocol massages are given in upper ease throughout
117	The names of abstract BTP protocol messages are given in upper-case throughout:
	BEGIN
118	
119	CONTEXT
120	RESIGN
121	
122	The values of elements within a BTP protocol message are indicated thus:
123	DECIDIA -
124	BEGIN/atom
125	
126	BTP protocol messages that are related semantically are joined by an ampersand:
127	
128	BEGIN/atom & CONTEXT
129	
130	BTP protocol messages that are transmitted together in a compound are joined by a + sign:
131	
132	ENROL + VOTE
133	
134	XML schemata and instances are given in Courier:
135	
136 137	<pre><btp:begin> </btp:begin></pre>
137	Illustrative fragments of code in other languages, such as Java, are given in Lucida Console:
138	mustrative fragments of code in other ranguages, such as Java, are given in Euclua Console.
140	int main (String[] args)
141	
142	{ }
143	
144	Terms such as MUST, MAY and so on, which are defined in RFC [TBD number], "[TBD
145	title]" are used with the meanings given in that document but are given in lowercase bold,
146	rather than in upper-case:

147	
148	An Inferior must send one of RESIGN, PREPARED or CANCELLED to its
149	Superior.
150	-
151	

151 **Contents**

152		
153	Copyright and related notices	
154	Acknowledgements	
155	Typographical and Linguistic Conventions and Style	
156	Contents	
157	Part 1. Purpose and Features of BTP	
158	Introduction	
159	Development and Maintenance of the Specification	
160	Overview of the Business Transaction Protocol	12
161	Part 2. Normative Specification of BTP	15
162	Actors, Roles and Relationships	15
163	Relationships	15
164	Roles involved in the outcome relationships	17
165	Superior	17
166	Inferior	
167	Enroller	
168	Participant	20
169	Sub-coordinator	20
170	Sub-composer	
171	Roles involved in the control relationships	
172	Decider	
173	Coordinator	
174	Composer	
175	Terminator	
176	Initiator	
177	Factory	
178	Other roles	
179	Redirector	
180	Status Requestor	
181	Abstract Messages and Associated Contracts	
182	Addresses	
183	Request/response pairs	
184	Compounding messages	
185	Extensibility	
186	Messages	
187	Qualifiers	
188	Messages not restricted to outcome or control relationships	
189	CONTEXT	
190	CONTEXT_REPLY	
191	REQUEST_STATUS	
192	STATUS	
193	FAULT	
194	REQUEST_INFERIOR_STATUSES, INFERIOR_STATUSES	
195	Messages used in the outcome relationships	
196	ENROL	

197	ENROLLED	40
198	RESIGN	41
199	RESIGNED	42
200	PREPARE	43
201	PREPARED	43
202	CONFIRM	45
203	CONFIRMED	46
204	CANCEL	47
205	CANCELLED	48
206	CONFIRM_ONE_PHASE	49
207	HAZARD	50
208	CONTRADICTION	51
209	SUPERIOR_STATE	51
210	INFERIOR_STATE	53
211	REDIRECT	55
212	Messages used in control relationships	
213	BEGIN	56
214	BEGUN	57
215	PREPARE_INFERIORS	58
216	CONFIRM_TRANSACTION	59
217	TRANSACTION_CONFIRMED	61
218	CANCEL_TRANSACTION	
219	CANCEL_INFERIORS	
220	TRANSACTION_CANCELLED	
221	REQUEST_INFERIOR_STATUSES	
222	INFERIOR_STATUSES	
223	Groups – combinations of related messages	
224	CONTEXT & application message	
225	CONTEXT_REPLY & ENROL	
226	CONTEXT_REPLY (& ENROL) & PREPARED / & CANCELLED	
227	CONTEXT_REPLY & ENROL & application message (& PREPARED)	
228	BEGUN & CONTEXT	
229	BEGIN & CONTEXT	
230	Standard qualifiers	
231	Transaction timelimit	
232	Inferior timeout	
233	Minimum inferior timeout	
234	Inferior name	
235	State Tables	
236	Explanation of the state tables	
237	Status queries	
238	Decision events	
239	Disruptions – failure events	
240	Invalid cells and assumptions of the communication mechanism	
241	Meaning of state table events	
242	Persistent information	
243	Failure Recovery	

245 Persistent information 99 246 Redirection 99 247 Terminator:Decider failures 99 248 XML representation of Message Set 99 249 Addresses 99 250 Qualifiers 99 251 Identifiers 99 251 Identifiers 90 253 Messages 100 254 CONTEXT_REPLY 100 255 CONTEXT_REPLY 100 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 250 ENROL 10 251 ENROL 10 252 MESSAGENE 10 254 CONTEXT_REPLY 10 255 CONTEXT_REPLY 10 256 REQUEST_ATUS 10 257 STATUS 10 258 FAULT 10 260 ENROL 10 261 RESIGN 10 262	244	Types of failure	
246 Redirection	245		
247 Terminator:Decider failures 99 248 XML representation of Message Set 99 249 Addresses 99 250 Qualifiers 99 251 Identifiers 100 252 Message References 100 253 Messages 100 254 CONTEXT_REPLY 100 255 CONTEXT_REPLY 100 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRM 10 267 CANCEL 10 268 FAULT 10 269 CONFIRM 10 260 CONFIRM 10 261 RESIGN	246		
248 XML representation of Message Set	247		
249 Addresses 99 250 Qualifiers 99 251 Identifiers 100 252 Message References 100 253 Messages 100 254 CONTEXT 100 255 CONTEXT_REPLY 100 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRM 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRMED 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 274 REDIRECT 10 <tr< td=""><td>248</td><td></td><td></td></tr<>	248		
250 Qualifiers 99 251 Identifiers 10 252 Message References 10 253 Messages 10 254 CONTEXT 10 255 CONTEXT_REPLY 10 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARE 10 265 CONFIRM 10 266 CONFIRM 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM 10 266 CONFIRM 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10	-		
251 Identifiers 10 252 Message References 10 253 Messages 10 254 CONTEXT 10 255 CONTEXT_REPLY 10 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 260 ENROLLED 10 261 RESIGNED 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRM 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10	-		
252 Message References 10 253 Messages 10 254 CONTEXT 10 255 CONTEXT_REPLY 10 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 260 ENROL 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARE 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10 277 PREPARE_INFERIORS 11			
253 Messages 100 254 CONTEXT 100 255 CONTEXT_REPLY 100 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 250 ENROL 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARE 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 100 269 CONFIRMED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGIN 10 <t< td=""><td>-</td><td></td><td></td></t<>	-		
254 CONTEXT 10 255 CONTEXT_REPLY 10 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARE 10 265 CONFIRM 10 266 CONFIRMED 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGIN 10 277 PREPARE_INFERIORS 11	253		
255 CONTEXT_REPLY 10 256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 250 ENROL 10 260 ENROLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARE 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONFIRM_ONE_PHASE 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGIN 10 277 PREPARE_INFERIORS 11 278 CONFIRM_TRANSACTION 11 <td></td> <td></td> <td></td>			
256 REQUEST_STATUS 10 257 STATUS 10 258 FAULT 10 259 ENROL 10 260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARE 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONFIRM_ONE_PHASE 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10 277 PREPARE_INFERIORS 11 278 CONFIRMED 10 274 REDURECT 10 275 BEGIN 10	255		
257 STATUS 10 258 FAULT 10 259 ENROL 10 260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10 277 PREPARE_INFERIORS 11 278 CONFIRM_TRANSACTION 10 279 TRANSACTION_CONFIRMED 11 280 CANCEL_INFERIORS 11 281 CANCEL_TRANSACTION	256		
258 FAULT. 10 259 ENROL 10 260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10 277 PREPARE_INFERIORS 11 278 CONFIRM_TRANSACTION 11 279 TRANSACTION_CONFIRMED 11 280 CANCEL_TRANSACTION 11 281 CANCEL_INFERIORS 11 282 TRANSACTION_CONF	257		
259 ENROL 10 260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRMED 10 266 CONFIRMED 10 266 CONFIRMONE_PHASE 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10 277 PREPARE_INFERIORS 11 278 CONFIRM_TRANSACTION 11 279 TRANSACTION_CONFIRMED 110 270 TRANSACTION_CONFIRMED 11 280 CANCEL_IN	258		
260 ENROLLED 10 261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10 277 PREPARE_INFERIORS 11 278 CONFIRM_TRANSACTION 10 279 TRANSACTION_CONFIRMED 11 280 CANCEL_INFERIORS 11 281 CANCEL_INFERIORS 11 282 TRANSACTION 11 284 INFERIOR_STATUSES 11 285 S	259	ENROL	
261 RESIGN 10 262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGIN 10 277 PREPARE_INFERIORS 11 278 CONFIRM_TRANSACTION 11 279 TRANSACTION_CONFIRMED 11 280 CANCEL_INFERIORS 11 281 CANCEL_INFERIORS 11 282 TRANSACTION_CANCELLED 11 283 REQUEST_INFERIOR_STATUSES 11 284 INFERIOR_STATUSES 11	260		
262 RESIGNED 10 263 PREPARE 10 264 PREPARED 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGIN 10 277 PREPARE_INFERIORS 10 278 CONFIRM_TRANSACTION 11 279 TRANSACTION_CONFIRMED 11 280 CANCEL_TRANSACTION 11 281 CANCEL_INFERIORS 11 282 TRANSACTION_CANCELLED 11 283 REQUEST_INFERIOR_STATUSES 11 284 INFERIOR_STATUSES 11 285 Standard qualifiers 11 <td>261</td> <td></td> <td></td>	261		
263 PREPARE 100 264 PREPARED 100 265 CONFIRM 100 266 CONFIRMED 100 267 CANCEL 100 268 CANCELLED 100 269 CONFIRM_ONE_PHASE 100 270 HAZARD 100 271 CONTRADICTION 100 272 SUPERIOR_STATE 100 273 INFERIOR_STATE 100 274 REDIRECT 100 275 BEGIN 100 276 BEGUN 100 277 PREPARE_INFERIORS 110 278 CONFIRM_TRANSACTION 110 279 TRANSACTION_CONFIRMED 110 279 TRANSACTION_CONFIRMED 111 280 CANCEL_TRANSACTION 111 281 CANCEL_TRANSACTION 111 282 TRANSACTION_CANCELLED 111 283 REQUEST_INFERIOR_STATUSES 111 284 INFERIOR_STATUSES 111 285 Standard qua	262		
264 PREPARED 10 265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10 277 PREPARE_INFERIORS 10 277 PREPARE_INFERIORS 11 278 CONFIRMED 10 279 TRANSACTION_CONFIRMED 11 280 CANCEL_TRANSACTION 11 281 CANCEL_INFERIORS 11 282 TRANSACTION_CANCELLED 11 283 REQUEST_INFERIOR_STATUSES 11 284 INFERIOR_STATUSES 11 285 Standard qualifiers 11	263		
265 CONFIRM 10 266 CONFIRMED 10 267 CANCEL 10 268 CANCELLED 10 269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 10 273 INFERIOR_STATE 10 274 REDIRECT 10 275 BEGIN 10 276 BEGUN 10 277 PREPARE_INFERIORS 11 278 CONFIRM_TRANSACTION 11 279 TRANSACTION_CONFIRMED 11 279 TRANSACTION_CONFIRMED 11 280 CANCEL_TRANSACTION 11 281 CANCEL_INFERIORS 11 282 TRANSACTION_CANCELLED 11 283 REQUEST_INFERIOR_STATUSES 11 284 INFERIOR_STATUSES 11 285 Standard qualifiers 11	264		
266 CONFIRMED 100 267 CANCEL 100 268 CANCELLED 100 269 CONFIRM_ONE_PHASE 100 270 HAZARD 100 271 CONTRADICTION 100 272 SUPERIOR_STATE 100 273 INFERIOR_STATE 100 274 REDIRECT 100 275 BEGIN 100 276 BEGUN 100 277 PREPARE_INFERIORS 100 277 PREPARE_INFERIORS 110 278 CONFIRM_TRANSACTION 110 279 TRANSACTION_CONFIRMED 110 279 TRANSACTION_CONFIRMED 111 280 CANCEL_TRANSACTION 111 281 CANCEL_INFERIORS 111 282 TRANSACTION_CANCELLED 111 283 REQUEST_INFERIOR_STATUSES 111 284 INFERIOR_STATUSES 111 285 Standard qualifiers 111	265		
267 CANCEL 100 268 CANCELLED 100 269 CONFIRM_ONE_PHASE 100 270 HAZARD 100 271 CONTRADICTION 100 272 SUPERIOR_STATE 100 273 INFERIOR_STATE 100 274 REDIRECT 100 275 BEGIN 100 276 BEGUN 100 277 PREPARE_INFERIORS 100 277 PREPARE_INFERIORS 110 278 CONFIRM_TRANSACTION 110 279 TRANSACTION_CONFIRMED 110 279 TRANSACTION_CONFIRMED 111 280 CANCEL_TRANSACTION 111 281 CANCEL_INFERIORS 111 282 TRANSACTION_CANCELLED 111 283 REQUEST_INFERIOR_STATUSES 111 284 INFERIOR_STATUSES 111 285 Standard qualifiers 111	266		
269 CONFIRM_ONE_PHASE 10 270 HAZARD 10 271 CONTRADICTION 10 272 SUPERIOR_STATE 103 273 INFERIOR_STATE 103 274 REDIRECT 103 275 BEGIN 100 276 BEGUN 100 277 PREPARE_INFERIORS 100 278 CONFIRM_TRANSACTION 110 279 TRANSACTION_CONFIRMED 110 280 CANCEL_TRANSACTION 11 281 CANCEL_INFERIORS 11 282 TRANSACTION_CANCELLED 11 283 REQUEST_INFERIOR_STATUSES 112 284 INFERIOR_STATUSES 112 285 Standard qualifiers 111	267	CANCEL	
270 HAZARD 10 271 CONTRADICTION. 10 272 SUPERIOR_STATE. 10 273 INFERIOR_STATE. 10 274 REDIRECT. 10 275 BEGIN. 10 276 BEGUN. 10 277 PREPARE_INFERIORS 10 278 CONFIRM_TRANSACTION 11 279 TRANSACTION_CONFIRMED. 11 280 CANCEL_TRANSACTION 11 281 CANCEL_INFERIORS 11 282 TRANSACTION_CANCELLED. 11 283 REQUEST_INFERIOR_STATUSES 11 284 INFERIOR_STATUSES 11 285 Standard qualifiers 11	268		
270 HAZARD 10 271 CONTRADICTION. 10 272 SUPERIOR_STATE. 10 273 INFERIOR_STATE. 10 274 REDIRECT. 10 275 BEGIN. 10 276 BEGUN. 10 277 PREPARE_INFERIORS 10 278 CONFIRM_TRANSACTION 11 279 TRANSACTION_CONFIRMED. 11 280 CANCEL_TRANSACTION 11 281 CANCEL_INFERIORS 11 282 TRANSACTION_CANCELLED. 11 283 REQUEST_INFERIOR_STATUSES 11 284 INFERIOR_STATUSES 11 285 Standard qualifiers 11	269	CONFIRM ONE PHASE	
271 CONTRADICTION. 10 272 SUPERIOR_STATE. 103 273 INFERIOR_STATE. 103 274 REDIRECT. 103 275 BEGIN 100 276 BEGUN. 100 277 PREPARE_INFERIORS 110 278 CONFIRM_TRANSACTION 110 279 TRANSACTION_CONFIRMED. 110 280 CANCEL_TRANSACTION 111 281 CANCEL_INFERIORS 111 282 TRANSACTION_CANCELLED. 111 283 REQUEST_INFERIOR_STATUSES 111 284 INFERIOR_STATUSES 111 285 Standard qualifiers 111	270		
272 SUPERIOR_STATE	271		
273 INFERIOR_STATE	272		
274 REDIRECT	273		
276 BEGUN	274		
277PREPARE_INFERIORS110278CONFIRM_TRANSACTION110279TRANSACTION_CONFIRMED110280CANCEL_TRANSACTION111281CANCEL_INFERIORS111282TRANSACTION_CANCELLED111283REQUEST_INFERIOR_STATUSES111284INFERIOR_STATUSES111285Standard qualifiers111	275	BEGIN	
278CONFIRM_TRANSACTION110279TRANSACTION_CONFIRMED110280CANCEL_TRANSACTION111281CANCEL_INFERIORS111282TRANSACTION_CANCELLED111283REQUEST_INFERIOR_STATUSES112284INFERIOR_STATUSES112285Standard qualifiers112	276	BEGUN	
279TRANSACTION_CONFIRMED.110280CANCEL_TRANSACTION11281CANCEL_INFERIORS11282TRANSACTION_CANCELLED.11283REQUEST_INFERIOR_STATUSES112284INFERIOR_STATUSES112285Standard qualifiers112	277	PREPARE_INFERIORS	
280CANCEL_TRANSACTION11281CANCEL_INFERIORS11282TRANSACTION_CANCELLED11283REQUEST_INFERIOR_STATUSES112284INFERIOR_STATUSES112285Standard qualifiers112	278	CONFIRM_TRANSACTION	
281CANCEL_INFERIORS11282TRANSACTION_CANCELLED11283REQUEST_INFERIOR_STATUSES112284INFERIOR_STATUSES112285Standard qualifiers112	279	TRANSACTION_CONFIRMED	
282TRANSACTION_CANCELLED.11283REQUEST_INFERIOR_STATUSES112284INFERIOR_STATUSES112285Standard qualifiers112	280	CANCEL_TRANSACTION	
283REQUEST_INFERIOR_STATUSES112284INFERIOR_STATUSES112285Standard qualifiers112	281	CANCEL_INFERIORS	
284 INFERIOR_STATUSES 112 285 Standard qualifiers 112	282	TRANSACTION_CANCELLED	
285 Standard qualifiers	283	REQUEST_INFERIOR_STATUSES	
1	284	INFERIOR_STATUSES	
	285	Standard qualifiers	
286 Transaction timelimit11	286	Transaction timelimit	
287 Inferior timeout	287	Inferior timeout	
288 Minimum inferior timeout	288	Minimum inferior timeout	
289 Inferior name11:	289		
290 Compounding of Messages11:	290	Compounding of Messages	

291	XML Schemas	
292	XML schema for BTP messages	117
293	XML schema for standard qualifiers	
294	Carrier Protocol Bindings	
295	Carrier Protocol Binding Proforma	
296	Bindings for request/response carrier protocols	
297	Request/response exploitation rules	
298	SOAP Binding	
299	Example scenario using SOAP binding	
300	SOAP + Attachments Binding	
301	Conformance	141
302	Part 3. Appendices	144
303	A. Glossary	144
304		

³⁰⁵ Part 1. Purpose and Features of BTP

306

307 Introduction

308 309

310

311

312

313

314

318

323

331

337

This document, which describes and defines the Business Transaction Protocol (BTP), is a Committee Specification of the Organization for the Advancement of Structured Information Standards (OASIS). The standard has been authored by the collective work of representatives of ten software product companies (listed on page 3), grouped in the Business Transactions Technical Committee (BT TC) of OASIS.

The OASIS BTP Technical Committee began its work at an inaugural meeting in San Jose,
Calif. on 13 March 2001, and this specification was endorsed as a Committee Specification
by a [*** unanimous] vote on [*** date].

BTP uses a two-phase outcome coordination protocol to create atomic effects (results of computations). BTP also permits the composition of such atomic units of work (atoms) into cohesive business transactions (cohesions), which allow application intervention into the selection of the atoms which will be confirmed, and of those which will be cancelled.

BTP is designed to allow transactional coordination of participants, which are part of services
offered by multiple autonomous organizations (as well as within a single organization). It is
therefore ideally suited for use in a Web Services environment. For this reason this
specification defines communications protocol bindings which target the emerging Web
Services arena, while preserving the capacity to carry BTP messages over other
communication protocols. Protocol message structure and content constraints are schematized
in XML, and message content is encoded in XML instances.

The BTP allows great flexibility in the implementation of business transaction participants. Such participants enable the consistent reversal of the effects of atoms. BTP participants may use recorded before- or after-images, or compensation operations to provide the "rollforward, roll-back" capacity which enables their subordination to the overall outcome of an atomic business transaction.

The BTP is an interoperation protocol which defines the roles which software agents (actors) may occupy, the messages that pass between such actors, and the obligations upon and commitments made by actors-in-roles. It does not define the programming interfaces to be used by application programmers to stimulate message flow or associated state changes.

The BTP is based on a permissive and minimal approach, where constraints on
implementation choices are avoided. The protocol also tries to avoid unnecessary
dependencies on other standards, with the aim of lowering the hurdle to implementation.

- 346
- 347
- 348

348	Development and Maintenance of the Specification
349	
350	For more information on the genesis and development of BTP, please consult the OASIS BT
351	Technical Committee's website, at
352	
353	http://www.oasis-open.org/committees/business-transactions/
354	
355	
356	As of the date of adoption of this specification the OASIS BT Technical Committee is still in
357	existence, with the charter of
358	maintaining the specification in the light of implementation experiences
359 360	
361 362	coordinating publicity for BTP
363	□ liaising with other standards bodies whose work affects or may be affected by
364 365	BTP
366	• reviewing the appropriate time, in the light of implementation experience and
367	user support, to put BTP forward for adoption as a full OASIS standard
368	
369	
370	If you have a question about the functionality of BTP, or wish to report an error or to suggest
371	a modification to the specification, please subscribe to:
372	
373	<u>bt-spec@lists.oasis-open.org</u>
374	
375	Any employee of a corporate member of OASIS, or any individual member of OASIS, may
376 377	subscribe to OASIS mail lists, and is also entitled to apply to join the Technical Committee.
378	The main list of the committee is:
379	The main list of the committee is.
380	business-transaction@lists.oasis-open.org
381	
382	
383	
384	
385	
386	

Overview of the Business Transaction Protocol

387

391

401

413

414

415 416

417

418

A Business Transaction is a consistent change in the state of a business relationship between
 two or more parties. BTP provides means to allow the consistent and coordinated changes in
 the relationship as viewed from each party.

BTP assumes that for a given business transaction state changes occur, or are desired, in some set of parties, and that these changes are related in some business-defined manner.
 394

Typically business-defined messages ("application messages") are exchanged between the parties to the transaction, which result in the performance of some set of operations. These operations create provisional or tentative state changes (the transaction's effect). The provisional changes of each party must either be confirmed (given final effect), or must be cancelled (counter-effected). Those parties which are confirmed create an atomic unit, within which the business transaction should have a consistent final effect.

The meaning of "effect", "final effect" and "counter-effect" is specific to each business transaction and to each party's role within it. A party may log intended changes (as its effect) and only process them as visible state changes on confirmation (its final effect). Or it may make visible state changes and store the information needed to cancel (its effect), and then simply delete the information needed for cancellation (its final effect). A counter-effect may be a precise inversion or removal of provisional changes, or it may be the processing of operations that in some way compensate for, make good, alleviate or supplement their effect.

To ensure that confirmation or cancellation of the provisional effect within different parties
can be consistently performed, it is necessary that each party should

- determine whether it is able both to cancel (counter-effect) and to confirm (give final effect to) its effect
 - report its ability or inability to cancel-or-confirm (its preparedness) to a central coordinating entity

After receiving these reports, the coordinating entity is responsible for determining which of
the parties should be instructed to confirm and which should be instructed to cancel.

Such a two-phase exchange (ask, instruct) mediated by a central coordinator is required to
achieve a consistent outcome for a set of operations. BTP defines the means for software
agents executing on network nodes to interoperate using a two-phase coordination protocol,
leading either to the abandonment of the entire attempted transaction, or to the selection of an
internally consistent set of confirmed operations.

BTP centres on the bilateral relationship between the computer systems of the coordinating entity and those of one of the parties in the overall business transaction. In that relationship a software agent within the coordinating entity's systems plays the BTP role of Superior for a given transaction and one or more software agents within the systems of the party play the BTP role of Inferior. Each Inferior has one Superior, therefore, while a single Superior may have multiple Inferiors within each party to the transaction, and may be related to Inferiors
within multiple parties. Each Superior:Inferior pair exchanges protocol-defined messages.

- 436 An Inferior is associated with some set of operation invocations that creates effect (provisional or tentative changes) within the party, for a given business transaction. The 437 438 Inferior is responsible for reporting to its related Superior whether its associated operations' 439 effect can be confirmed/cancelled. A Superior is responsible for gathering the reports of all of 440 its Inferiors, in order to ascertain which should be cancelled or confirmed. For example, if a Superior is acting as an atomic Coordinator it will treat any Inferior which cannot prepare to 441 442 cancel/confirm as having veto power over the whole business transaction, causing the 443 Superior to instruct all its Inferiors to cancel. A Superior may, under the dictates of a 444 controlling application, increase or reduce the set of Inferiors to which a common confirm or 445 cancel outcome may be delivered. Thus, the set of prepared Inferiors may be larger than the set of confirmed Inferiors. 446
- 447

435

448 An Inferior: Superior relationship is typically established in relation to one or more 449 application messages sent from one part of the application (linked to the Superior) to some other part of the application to request the performance of operations that are to be subject to 450 the confirm or cancel decision of the Superior. If an application is divided between a client 451 452 and a service, which use RPCs to communicate application requests and responses, then the 453 client would typically be associated with the Superior and the service would typically host the 454 Inferior(s). (BTP does not mandate such an application topology nor does it require the use of 455 RPC or any other application communication paradigm.)

456

BTP defines a CONTEXT message that can be sent "in relation to" such application 457 messages. On receipt of a CONTEXT, one or more Inferiors may be created and "enrolled" 458 459 with the Superior, establishing the Superior:Inferior relationships. The particular mechanisms by which a CONTEXT is "related" to application messages is an issue for the application 460 protocol and its binding to carrier mechanisms. BTP does not require that the enrolment is 461 requested by any particular entity – in a particular implementation this may be done by the 462 463 Inferior itself, by parts of the application or by other entities involved in the transmission of 464 the CONTEXT and the application messages. BTP defines a CONTEXT REPLY message that can be sent on the return path of the CONTEXT to indicate whether the enrolment was 465 successful. Without CONTEXT REPLY it would be possible for a Superior to have an 466 incorrect view of which Inferiors it was supposed to involve in its confirm decision. 467 468

- 469 It should be noted that this BTP specification recognises that:
- an Inferior may itself be a Superior to other BTP Inferiors; this occurs when some of
 the operations associated with the Inferior involve other application elements whose
 operations are to be subject to the confirm/cancel instruction sent to the Inferior. The
 specification treats any lower Inferiors as part of the associated operations;
- the requirement on an Inferior to be able to confirm or cancel does not include any specific mechanism to determine the isolation of the effects of operations; the requirement is only that the Inferior is able to confirm or cancel the operations, as their effects are known to the Superior and the application directly in contact with the Superior. Thus the confirm-or-cancel requirement may be achieved by performing all the operations and remembering a compensating counter operation (that will be

480	triggered by a cancel order); or by remembering the operations (having checked they
481	are valid) and performing them only if a confirm order is received; or by forbidding
482	any other access to data changed by the operations and releasing them in their
483	unchanged state (if cancelled) or their changed state (if confirmed); or by various
484	combinations of these. In addition, a cancellation may not return data to their original
485	state, but only to a state accepted by the application as appropriate to a cancelled
486	operation.
487	
488	
489	
490	
/01	

⁴⁹³ Part 2. Normative Specification of BTP

494

496

506

512

516

521

526 527

528

529

531

495 Actors, Roles and Relationships

497 Actors are software agents which process computations. BTP actors are addressable for the
498 purposes of receiving application and BTP protocol messages transmitted over some
499 underlying communications or carrier protocol. (See section "Addressing" for more detail.)
500

501 BTP actors play roles in the sending, receiving and processing of messages. These roles are 502 associated with responsibilities or obligations under the terms of software contracts defined 503 by this specification. (These contracts are stated formally in the sections entitled "Abstract 504 Messages and Associated Contracts" and "State Tables".) A BTP actor's computations put 505 the contracts into effect.

A role is defined and described in terms of a single business transaction. An implementation
supporting a role may, as an addressable entity, play the same role in multiple business
transactions, simultaneously or consecutively, or a separate addressable entity may be created
for each transaction. This is a choice for the implementer, and the addressing mechanisms
allow interoperation between implementations that make different choices.

513 Within a single transaction, one actor may play several roles, or each role may be assigned to 514 a distinct actor. This is again a choice for the implementer. An actor playing a role is termed 515 an "actor-in-role".

Actors may interoperate, in the sense that the roles played by actors may be implemented
using software created by different vendors for each actor-in-role. The section
"Conformance", gives guidelines on the groups of roles that may be implemented in a
partial, interoperable implementation of BTP.

522The descriptions of the roles concentrate on the normal progression of a business transaction,523and some of the more important divergences from this. They do not cover all exception cases524- the message set definition and the state tables provide a more comprehensive specification.525

Note – A BTP role is approximately equivalent to an interface in some distributed computing mechanisms, or a port-type in WSDL. The definition of a role includes behaviour.

530 Relationships

- 532 There are two primary relationships in BTP.
- 533Image: Between an application element that determines that a business transaction should be
completed (the role of Terminator) and the BTP actor at the top of the transaction tree
(the role of Decider);

536	
537 538 539	 Between BTP actors within the tree, where one (the Superior) will inform the other (the Inferior) what the outcome decision is.
540 541 542	These primary relationships are involved in arriving at a decision on the outcome of a business transaction, and propagating that decision to all parties to the transaction. Taking the path that is followed when a business transaction is confirmed:
543 544	1. The Terminator determines that the business transaction should confirm, if it can; or (for a Cohesion), which parts should confirm
545 546	2. The Terminator asks the Decider to apply the desired outcome to the tree, if it can guarantee the consistency of the confirm decision
547 548	3. The Decider, which is Superior to one or more Inferiors, asks its Inferiors if they can agree to a confirm decision (for a Cohesion, this may not be all the Inferiors)
549 550	4. If any of those Inferiors are also Superiors, they ask their Inferiors and so on down the tree
551	5. Inferiors that are not Superiors report if they can agree to a confirm to their Superior
552 553	6. Inferiors that are also Superiors report their agreement only if they received such agreement from their Inferiors, and can agree themselves
554 555 556 557	 Eventually agreement (or not) is reported to the Decider. If all have agreed, the Decider makes and persists the confirm decision (hence the term "Decider" – it decides, everything else just asked); if any have disagreed, or if the confirm decision cannot be persisted, a cancel decision is made
558	8. The Decider, as Superior tells its Inferiors of the outcome
559	9. Inferiors that are also Superiors tell their Inferiors, recursively down the tree
560 561 562	10. The Decider replies to the Terminator's request to confirm, reporting the outcome decision
562 563 564 565 566 567 568	There are other relationships that are secondary to Terminator:Decider, Superior:Inferior, mostly involved in the establishment of the primary relationships. The various particular relationships can be grouped as the "control" relationships – primarily Terminator:Decider, but also Initiator:Factory; and the "outcome" relationships – primarily Superior:Inferior, but also Enroller:Superior.
569 570 571 572	The two groups of relationships are linked in that a Decider is a Superior to one or more Inferiors. There are also similarities in the semantics of some of the exchanges (messages) within the relationships. However they differ in that
573 574 575 576	1. All exchanges between Terminator and Decider are initiated by the Terminator (it is essentially a request/response relationship); either of Superior or Inferior may initiate messages to the other

- 2. The Superior:Inferior relationship is recoverable depending on the progress of the relationship, the two sides will re-establish their shared state after failure; the Terminator:Decider relationship is not recoverable
 - 3. The nature of the Superior:Inferior relationship requires that the two parties know of each other's addresses from when the relationship is established; the Decider does not need to know the address of the Terminator (provided it has some way of returning the response to a received message).

586 In the following sections, the responsibility of each role is defined, and the messages that are 587 sent or received by that role are listed. Note that some roles exist only to have a name for an 588 actor that issues a message and receives a reply to that message. Some of these roles may be 589 played by several actors in the course of a single business transaction.

591 **Roles involved in the outcome relationships**

593 Superior

577

578 579

580 581

582

583

584

585

590

592

594

604

612

617 618 619

623

595 Accepts enrolments from Inferiors, establishing a Superior: Inferior relationship with each. In 596 cooperation with other actors and constrained by the messages exchanged with the Inferior, 597 the Superior determines the **Outcome** applicable to the Inferior and informs the Inferior by 598 sending CONFIRM or CANCEL. This outcome can be confirm only if a PREPARED 599 message is received from the Inferior, and if a record, identifying the Inferior can be 600 persisted. (Whether this record is also a record of a confirm decision depends on the Superior's position in the business transaction as a whole.). The Superior must retain this 601 602 persistent record until it receives a CONFIRMED (or, in exceptional cases, CANCELLED or 603 HAZARD) from the Inferior.

A Superior may delegate the taking of the confirm or cancel decision to an Inferior, if there is
 only one Inferior, by sending CONFIRM_ONE_PHASE.

A Superior may be *Atomic* or *Cohesive;* an Atomic Superior will apply the same decision to
all of its Inferiors; a Cohesive Superior may apply confirm to some Inferiors and cancel to
others, or may confirm some after others have reported cancellation. The set of Inferiors that
the Superior confirms (or attempts to confirm) is called the "confirm-set".

- If RESIGN is received from an Inferior, the Superior:Inferior relationship is ended; the
 Inferior has no further effect on the behaviour of the Superior as a whole.
- 616 A Superior receives

ENROL

- 620 to enrol a new Inferior, establishing a new Superior:Inferior relationship.
- 621622 A Superior sends

624	ENROLLED
625 626	in reply to ENROL, if the appropriate parameter on the ENROL asked for the reply.
620 627	
628	A Superior sends
629	
630 621	PREPARE
631 632	CONFIRM CANCEL
633	RESIGNED
634	CONFIRM_ONE_PHASE
635	SUPERIOR_STATE
636	_
637	to an enrolled Inferior.
638	
639	A Superior receives
640	
641 642	PREPARED
642 643	CANCELLED CONFIRMED
643 644	HAZARD
645	RESIGN
646	INFERIOR_STATE
647	
648	from an enrolled Inferior.
649	
650	Inferior
651	
652	Responsible for applying the Outcome to some set of associated operations – the application
653	determines which operations are the responsibility of a particular Inferior.
654 655	An Inferior is Enrolled with a single Superior (hereafter referred to as "its Superior"),
656	establishing a Superior: Inferior relationship. If the Inferior is able to ensure that either a
657	confirm or cancel decision can be applied to the associated operations, and can persist
658	information to retain that condition, it sends a PREPARED message to the Superior. When
659	the Outcome is received from the Superior, the Inferior applies it, deletes the persistent
660	information, and replies with CANCELLED or CONFIRMED as appropriate.
661	
662	If an Inferior is unable to come to a prepared state, it cancels the associated operations and
663	informs the Superior with a CANCELLED message. If it is unable to either come to a
664 665	prepared state, or to cancel the associated operations, it informs the Superior with a
665 666	HAZARD message.
667	An Inferior that has become prepared may, exceptionally, make an autonomous decision to be
668	applied to the associated operations, without waiting for the Outcome from the Superior. It is
669	required to persist this autonomous decision and report it to the Superior with CONFIRMED
670	or CANCELLED as appropriate. If, when CONFIRM or CANCEL is received, the

671	autonomous decision and the decision received from the Superior are contradictory, the		
672	Inferior must retain the record of the autonomous decision until receiving a		
673	CONTRADICTION message.		
674			
675	An Inferior receives		
676			
677	PREPARE		
678	CONFIRM		
679	CANCEL		
680	RESIGNED		
681	CONFIRM_ONE_PHASE		
682	SUPERIOR_STATE		
683			
	from its Superior		
684	from its Superior.		
685			
686	An Inferior sends		
687			
688	PREPARED		
689			
	CANCELLED		
690	CONFIRMED		
691	HAZARD		
692	RESIGN		
693	INFERIOR_STATE		
694	IN EARON_STATE		
695	to its Superior.		
()(
696			
696 697			
697	Enroller		
697 698	Enroller		
697 698 699			
697 698 699 700	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in		
697 698 699			
697 698 699 700 701	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some		
697 698 699 700 701 702	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a		
697 698 699 700 701 702 703	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving		
697 698 699 700 701 702 703 704	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a		
697 698 699 700 701 702 703 704 705	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT.		
697 698 699 700 701 702 703 704	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving		
697 698 699 700 701 702 703 704 705	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT.		
697 698 699 700 701 702 703 704 705 706 707	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends		
697 698 699 700 701 702 703 704 705 706 707 708	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT.		
697 698 699 700 701 702 703 704 705 706 707 708 709	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends		
697 698 699 700 701 702 703 704 705 706 707 708 709 710	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends		
697 698 699 700 701 702 703 704 705 706 707 708 709 710 711	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends ENROL to a Superior.		
697 698 699 700 701 702 703 704 705 706 707 708 709 710	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends		
697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends ENROL to a Superior.		
697 698 699 700 701 702 703 704 705 706 707 708 707 708 709 710 711 712 713	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends ENROL to a Superior. An Enroller receives		
697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends ENROL to a Superior.		
697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends ENROL to a Superior. An Enroller receives ENROLLED		
697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends ENROL to a Superior. An Enroller receives		
697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715	Causes the enrolment of an Inferior with a Superior. This role is distinguished because in some implementations the enrolment request will be performed by the application, in some the application will ask the actor that will play the role of Inferior to enrol itself, and a Factory may enrol a new Inferior (which will also be Superior) as a result of receiving BEGIN&CONTEXT. An Enroller sends ENROL to a Superior. An Enroller receives ENROLLED		

718	An ENROL message sent from an Enroller that did not require an ENROLLED response may
719	be modified en route to the Superior by an intermediate actor to ask for an ENROLLED
720	response to be sent to the intermediate. (This may occur in the "one-shot" scenario, where an
721	ENROL/no-rsp-req is received in relation to a CONTEXT_REPLY/related; the receiver of
722	the CONTEXT_REPLY will need to ensure the enrolment is successful).

724 Participant

723

725

731

735

An Inferior which is specialized for the purposes of an application. Some application
operations are associated directly with the Participant, which is responsible for determining
whether a prepared condition is possible for them, and for applying the outcome. ("associated
directly" as opposed to involving another BTP Superior:Inferior relationship, in which this
actor is the Superior).

The associated operations may be performed by the actor that has the role of Participant, or
they may be performed by another actor, and only the confirm/cancel application is
performed by the Participant.

In either case, the Participant, as part of becoming prepared (i.e. before it can send
PREPARED to the Superior), will persist information allowing it apply a confirm decision to
the operations and to apply a cancel decision. The nature of this information depends on the
operations.

740	Note – Possible approaches are:
741 742 743 744	• The operations may be performed completely and the Participant persists information to perform counter-effect operations (compensating operations) to apply cancellation;
745 746 747	o The operations may be just checked and not performed at all; the Participant persists information to perform them to apply confirmation;
748 749 750	o The Participants persists the prior state of data affected by the operations and the operations are performed; the Participant restores the prior state to apply cancellation;
751 752	o As the previous, but other access to the affected data is forbidden until the decision is known
753	
754	Sub-coordinator
755	
756	An Inferior which is also an Atomic Superior.
757	
758 759	A sub-coordinator is the Inferior in one Superior:Inferior relationship and the Superior in one or more Superior:Inferior relationships.

 From the perspective of its Superior (the one the sub-coordinator is Inferior to), there is no difference between a sub-coordinator and any other Inferior. From this perspective, the "associated operations" of the sub-coordinator as an Inferior include the relationships with Inferiors. A sub-coordinator does not become prepared (and send PREPARED to its Superior) until unless it has received PREPARED (or RESIGN) from all its Inferiors. The outcome is propagated to all Inferiors. Sub-composer An Inferior which is also a Cohesive Superior.
 difference between a sub-coordinator and any other Inferior. From this perspective, the "associated operations" of the sub-coordinator as an Inferior include the relationships with Inferiors. A sub-coordinator does not become prepared (and send PREPARED to its Superior) until a unless it has received PREPARED (or RESIGN) from all its Inferiors. The outcome is propagated to all Inferiors. Sub-composer An Inferior which is also a Cohesive Superior.
 "associated operations" of the sub-coordinator as an Inferior include the relationships with Inferiors. A sub-coordinator does not become prepared (and send PREPARED to its Superior) until unless it has received PREPARED (or RESIGN) from all its Inferiors. The outcome is propagated to all Inferiors. Sub-composer An Inferior which is also a Cohesive Superior.
 764 Inferiors. 765 766 A sub-coordinator does not become prepared (and send PREPARED to its Superior) until a unless it has received PREPARED (or RESIGN) from all its Inferiors. The outcome is propagated to all Inferiors. 769 770 Sub-composer 771 772 An Inferior which is also a Cohesive Superior.
 765 766 A sub-coordinator does not become prepared (and send PREPARED to its Superior) until a unless it has received PREPARED (or RESIGN) from all its Inferiors. The outcome is propagated to all Inferiors. 769 770 Sub-composer 771 772 An Inferior which is also a Cohesive Superior.
 A sub-coordinator does not become prepared (and send PREPARED to its Superior) until a unless it has received PREPARED (or RESIGN) from all its Inferiors. The outcome is propagated to all Inferiors. Sub-composer An Inferior which is also a Cohesive Superior.
 unless it has received PREPARED (or RESIGN) from all its Inferiors. The outcome is propagated to all Inferiors. Sub-composer An Inferior which is also a Cohesive Superior.
 768 propagated to all Inferiors. 769 770 Sub-composer 771 772 An Inferior which is also a Cohesive Superior.
 769 770 Sub-composer 771 772 An Inferior which is also a Cohesive Superior.
 770 Sub-composer 771 772 An Inferior which is also a Cohesive Superior.
An Inferior which is also a Cohesive Superior.
An Inferior which is also a Cohesive Superior.
1
773
The Like a sub-coordinator, a sub-composer cannot be distinguished from any other Inferior from the transformation of the sub-coordinator.
the perspective of its Superior.
776
A sub-composer is similar to a sub-coordinator, except that the constraints linking the
different Inferiors concern only those Inferiors in the confirm-set. How the confirm-set is
controlled, and when, is not defined in this specification.
780
781 If the sub-composer is instructed to cancel, by receiving a CANCEL message from its
782 Superior, the cancellation is propagated to all its Inferiors.
783
784
785 Roles involved in the control relationships
785 Roles involved in the control relationships786
 785 Roles involved in the control relationships 786 787 Decider
 Roles involved in the control relationships Decider 788
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node
 Roles involved in the control relationships 786 787 787 788 789 789 A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome for the transaction tree and receives requests from a Terminator as to the desired outcome fo
 Roles involved in the control relationships 786 787 788 788 789 A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for the business transaction. If the Terminator asks the Decider to confirm the business transaction
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for the business transaction. If the Terminator asks the Decider to confirm the business transaction the responsibility of the Decider to finally take the confirm decision. The taking of the
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for the usiness transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for to business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for the business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from 795
 Roles involved in the control relationships Decider Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for the business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from 795 A Decider is instructed to cancel by receiving CANCEL_TRANSACTION.
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for the business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from A Decider is instructed to cancel by receiving CANCEL_TRANSACTION.
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for to business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from A Decider is instructed to cancel by receiving CANCEL_TRANSACTION. A Decider that is an Atomic Superior (all Inferiors will have the same outcome) is a
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for to business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from A Decider is instructed to cancel by receiving CANCEL_TRANSACTION. A Decider that is an Atomic Superior (all Inferiors will have the same outcome) is a Coordinator. A Decider that is a Cohesive Superior (some Inferiors may cancel, some
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for to business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from the becider is instructed to cancel by receiving CANCEL_TRANSACTION. A Decider that is an Atomic Superior (all Inferiors will have the same outcome) is a Coordinator. A Decider that is a Cohesive Superior (some Inferiors may cancel, some confirm) is a Cohesion.
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for to business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from the decider is instructed to cancel by receiving CANCEL_TRANSACTION. A Decider that is an Atomic Superior (all Inferiors will have the same outcome) is a Coordinator. A Decider that is a Cohesive Superior (some Inferiors may cancel, some confirm) is a Cohesion.
 Roles involved in the control relationships Decider A Superior that is not also the Inferior on a Superior:Inferior relationship. It is the top-node the transaction tree and receives requests from a Terminator as to the desired outcome for to business transaction. If the Terminator asks the Decider to confirm the business transaction is the responsibility of the Decider to finally take the confirm decision. The taking of the decision is synonymous with the persisting of information identifying the Inferiors that are be confirmed. An Inferior cannot be confirmed unless PREPARED has been received from the confirmed to cancel by receiving CANCEL_TRANSACTION. A Decider that is an Atomic Superior (all Inferiors will have the same outcome) is a Coordinator. A Decider that is a Cohesive Superior (some Inferiors may cancel, some confirm) is a Cohesion. All Deciders receive
785Roles involved in the control relationships786787789789789790791791792793793794795795796797798798799799790791792793794795795796797798798799799798799799798799799799790791792793794795795796797798798799799790790791792793794795795796797798798799799790790791792793794795795796797798798799790791792793794795795796797798798799790791<
785Roles involved in the control relationships786787788789789789789790791791792793793794795795796797797798798799799794795795796797798798799799798799799799790791792793794795795796797798798799799790791792793794795795796797798798799799790791792793794795795796797798798799799790791792793794795795796797798798799799790791792793794795<
785Roles involved in the control relationships786787789789789790791791792793793794795795796797798798799799790791792793794795795796797798798799799798799799798799799799790791792793794795795796797798798799799790790791792793794795795796797798798799799790790791792793794795795796797798798799790791792793794795795796797798798799790791<

807	All Deciders send
808	CONFIRM_COMPLETE
809	CANCEL_COMPLETE
810	INFERIOR_STATUSES
811	
812	
813	Coordinator
814	
815	A Decider that is an Atomic Superior. The same outcome decision will be applied to all
816	Inferiors (excluding any from which RESIGN is received).
817	monors (chordaning any nom when recorded to recorded).
818	PREPARED must be received from all remaining Inferiors for a confirm decision to be taken.
819	
820	A Coordinator must make a cancel decision if
821	it is instructed to cancel by the Terminator
822	if CANCELLED is received from any Inferior
823	if it is unable to persist a confirm decision
824	If it is unable to persist a commin decision
824 825	Composer
825 826	Composer
820 827	A Desider that is a Calculus Superior. If the Terminator requests confirmation of the
827	A Decider that is a Cohesive Superior. If the Terminator requests confirmation of the Cohesion, that request will determine the confirm-set of the Cohesion.
	Conesion, that request will determine the confirm-set of the Conesion.
829 820	DDEDADED must be received from all Inferiors in the confirm set (evoluting only from
830 831	PREPARED must be received from all Inferiors in the confirm-set (excluding any from
	which RESIGN is received) for a confirm decision to be taken.
832	A Composer must make a sensel desision (applying to all Inferiors) if
833	A Composer must make a cancel decision (applying to all Inferiors) if
834 825	it is instructed to cancel by the Terminator
835	if CANCELLED is received from any Inferior in the confirm-set
836	if it is unable to persist a confirm decision
837	A Composer may be asked to measure some or all of its Inferiors by reasiving
838	A Composer may be asked to prepare some or all of its Inferiors by receiving
839	PREPARE_INFERIORS. It issues PREPARE to any of those Inferiors from which none of
840	PREPARED, CANCELLED or RESIGN have been received, and replies to the
841	PREPARE_INFERIORS with INFERIOR_STATUSES.
842	
843	A Composer may be asked to cancel some of its Inferiors, but not itself, by receiving
844	CANCEL_INFERIORS.
845	
846	- · ·
847	Terminator
848	
849	Asks a Decider to confirm the business transaction, or instructs it to cancel all or (for a
850	Cohesion) part of the business transaction.
851	
852	All communications between Terminator and Decider are initiated by the Terminator. A
853	Terminator is usually an application element.

854	
855	A request to confirm is made by sending CONFIRM_TRANSACTION to the target Decider.
856	If the Decider is a Cohesion Composer, the Terminator may select which of the Composer's
857	Inferiors are to be included in the confirm-set. If the Decider is an Atom Coordinator, all
858	Inferiors are included. After applying the decision, the Decider replies with
859	CONFIRM_COMPLETE, CANCEL_COMPLETE or (in the case of problems)
860	INFERIOR_STATUSES.
861	-
862	A Terminator may ask a Composer (but not a Coordinator) to prepare some or all of its
863	Inferiors with PREPARE_INFERIORS. The Composer replies with
864	INFERIOR_STATUSES.
865	INTERIOR_STATUSES.
80 <i>5</i> 866	A Terminator may and CANCEL TDANGACTION to instruct the Desider to concel the
	A Terminator may send CANCEL_TRANSACTION to instruct the Decider to cancel the
867	whole business transaction., The Decider replies with CANCEL_COMPLETE if all Inferiors
868	cancel successfully, and with INFERIOR_STATUSES in the case of problems. If the
869	Decider is a Cohesion Composer, the Terminator may send CANCEL_INFERIORS to cancel
870	some of the Inferiors; the Decider always replies with INFERIOR_STATUSES.
871	
872	A Terminator may check the status of the Inferiors of the Decider by sending
873	REQUEST_INFERIOR_STATUSES. The Decider replies with INFERIOR_STATUSES.
874	
875	A Terminator sends
876	CONFIRM_TRANSACTION
877	CANCEL_TRANSACTION
878	CANCEL_INFERIORS
879	PREPARE_INFERIORS
880	REQUEST_INFERIOR_STATUSES
881	
882	A Terminator receives
883	CONFIRM_COMPLETE
884	CANCEL COMPLETE
885	INFERIOR STATUSES
885	INTERIOR_STATUSES
	Initiator
887	Initiator
888	
889	Requests a Factory to create a Superior – this will either be a Decider (representing a new
890	top-level business transaction) or a sub-coordinator or sub-composer to be the Inferior of an
891	existing business transaction.
892	
893	An Initiator sends
894	
895	BEGIN
896	BEGIN & CONTEXT
897	
898	to a Factory, and receives in reply
899	
900	BEGUN & CONTEXT

901	
901 902	Factory
903	Tactory
904	Creates Superiors and returns the CONTEXT for the new Superior. The following types of
905	Superior are created :
906	
907	Decider, which is either
908	Composer or
909	Coordinator
910	Sub-composer
911	Sub-coordinator
912	
913	A Factory receives
914	
915	BEGIN
916	BEGIN & CONTEXT
917	1 1' '4
918 010	and replies with
919 920	BEGUN & CONTEXT
921	beoon a context
922	If the BEGIN has no related CONTEXT, the Factory creates a Decider, either a Cohesion
923	Composer or an Atom Coordinator, as determined by the "superior type" parameter on the
924	BEGIN.
925	
926	If the BEGIN has a related CONTEXT, the new Superior is also enrolled as an Inferior of the
927	Superior identified by the CONTEXT. The new Superior is thus a sub-composer or sub-
928	coordinator, as determined by the "superior type" parameter on the BEGIN.
929	
930	
931	
932	Other roles
933	
934	Redirector
935	
936	Sends a REDIRECT message to inform any actor that an address previously supplied for
937 028	some other actor is no longer appropriate, and to supply a new address or set of addresses to
938 939	replace the old one.
939 940	A Redirector may send a REDIRECT message in response to receiving a message using the
941	old address, or may send a REDIRECT at its own initiative.
942	If a Superior moves from the superior-address in its CONTEXT, or an Inferior moves from
943	the inferior-address in the ENROL message, the implementation must ensure that a
944	Redirector catches any inbound messages using the old address and replies with a
945	REDIRECT message giving the new address. (Note that the inbound message may itself be a
946	REDIRECT message.)
947	

948 949	A Redirect	or may also be used to change the address of other BTP actors.
950	After recei	ving a REDIRECT message, the BTP actor must use the new address not the old
951		s failure prevents it updating its information.
952	one, uniess	randie prevents it updating its information.
	Statue Do	nuestor
953	Status Rec	luestoi
954		
955	•	nd receives the current status of a transaction tree node – any of an Inferior,
956	·	r Decider, or the current status of the nodes relationships with its Inferiors, if any.
957		Status Requestor has no responsibilities – it is just a name for where the
958	-	_STATUS and REQUEST_INFERIOR_STATUSES comes from
959	(REQUES	T_INFERIOR_STATUSES is also issued by a Terminator to a Decider).
960		
961	A Status R	equestor sends
962		•
963	RE	EQUEST_STATUS
964		EQUEST_INFERIOR_STATUSES
965		
966	and receive	20
967		۵۸ ۱۳
968	STAT	
969	INFER	RIOR_STATUSES
970		
971	in response).
972		
973		er of the request can refuse to provide the status information by replying with
974		atusRefused). The information returned in STATUS will always relate to the
975	transaction	tree node as a whole (e.g. as an Inferior, even if it is also a Superior).
976		
977	Abstract N	lessages and Associated Contracts
978		al Massacas and defined in this section in terms of the shotnest information that has
979		ol Messages are defined in this section in terms of the abstract information that has
980		nunicated. These abstract messages will be mapped to concrete messages
981	communica	ated by a particular carrier protocol (there can be several such mappings defined).
982		
983	The abstrac	ct message set and the associated state table assume the carrier protocol will
984		
985		deliver messages completely and correctly, or not at all (corrupted messages will
986		not be delivered);
987		
988		report some communication failures, but will not necessarily report all (i.e. not all
989		message deliveries are positively acknowledged within the carrier);
990		
991		sometimes deliver successive messages in a different order than they were sent;
992	-	
993	and	
994	und	
// 1		

1007

1009

1014

1023

• does not have built-in mechanisms to link a request and a response

997 Note that these assumptions would be met by a mapping to SMTP and more than met by
998 mappings to SOAP/HTTP.
999

However, when the abstract message set is mapped to a carrier protocol that provides a richer
service (e.g. reports all delivery failures, guarantees ordered delivery or offers a
request/response mechanism), the mapping can take advantage of these features. Typically in
such cases, some of the parameters of an abstract message will be implicit in the carrier
mechanisms, while the values of other parameters will be directly represented in transmitted
elements.

1008 Addresses

1010All of the messages except CONTEXT and CONTEXT_REPLY have a "target address"1011parameter and many also have other address parameters. These latter identify the desired1012target of other messages in the set. In all cases, the exact value will invariably have been1013originally determined by the implementation that is the target or desired future target.

1015 The detailed format of the address will depend on the particular carrier protocol, but at this abstract level is considered to have three parts. The first part, the "binding name", identifies 1016 the binding to a particular carrier protocol – some bindings are specified in this document, 1017 1018 others can be specified elsewhere. The second part of the address, the "binding address", is meaningful to the carrier protocol itself, which will use it for the communication (i.e. it will 1019 permit a message to be delivered to a receiver). The third part, "additional information", is 1020 not used or understood by the carrier protocol. The "additional information" may be a 1021 1022 structured value.

1024 When a message is actually transmitted, the "binding name" of the target address will identify 1025 which carrier protocol is in use and the "binding address" will identify the destination, as 1026 known to the carrier protocol. The entire binding address is considered to be "consumed" by the carrier protocol implementation. All of it may be used by the sending implementation, or 1027 1028 some of it may be transmitted in headers, or as part of a URL in the carrier protocol, but then used or consumed by the receiving implementation of the carrier protocol to direct the BTP 1029 1030 message to a BTP-aware entity (BTP-aware in that it is capable of interpreting the BTP 1031 messages). The "additional information" of the target address will be part of the BTP message itself and used in some way by the receiving BTP-aware entity (it could be used to 1032 route the message on to some other BTP entity). Thus, for the target address, only the 1033 "additional information" field is transmitted in the BTP message and the "additional 1034 information" is opaque to parties other than the recipient. 1035

1036

1037 For other addresses in BTP messages, all three components will be within the message.

1038

1039 All messages that concern a particular Superior:Inferior relationship have an identifier 1040 parameter for the target side as well as the <u>compound</u>-target address. This allows full

1041 flexibility for implementation choices – an implementation can:

10.40	
1042 1043	a) Use the same binding address and additional information for multiple business
1043	transactions, using the identifier parameter to locate the relevant state
1045	information;
1046	b) Use the same binding address for multiple business transactions and use the
1047	additional information to locate the information; or
1048	c) Use a different binding address for each business transaction.
1049	
1050	Which of these choices is used is opaque to the entity sending the message – both parts of the
1051	address and the identifier originated at the recipient of this message (and were transmitted as
1052	parameters of earlier messages in the opposite direction). In cases b) and c), the identifier is to
1053	some extent redundant, although interoperation requires that it always be present.
1054	
1055	BTP recovery requires that the state information for a Superior or Inferior is accessible after
1056	failure and that the peer can distinguish between temporary inaccessibility and the permanent
1057	non-existence of the state information. As is explained in "Redirection" below, BTP provides
1058	mechanisms – having a set of BTP addresses for some parameters, and the REDIRECT
1059	message – that make this possible, even if the recovered state information is on a different
1060	address to the original one (as may be the case if case c) above is used).
1061	
1062	
1063	Request/response pairs
1064	
1065	Many of the messages combine in pairs as a request and its response. However, in some cases
1066	the response message is sent without a triggering request, or as a possible response to more
1067	than one type of request. To allow for this, the abstract message set treats each message as
1068 1069	standalone; but where a request does expect a reply, a "reply-address" parameter will be present. For any message with a reply address parameter, in the case of certain errors, a
1009	FAULT message will be sent to the reply address instead of the expected reply.
1070	TAOLT message will be sent to the reply address instead of the expected reply.
1071	For messages which are specified as sent between Superior and Inferior, a FAULT message is
1072	sent to the peer.
1074	
1075	Compounding messages
1075	Compounding messages
1070	BTP messages may be sent in combination with each other, or with other (application)
1078	messages. There are two cases:
1079	
1080	a) Sending the messages together where the combination has semantic
1081	significance. One message is said to be "related to" the other – the combination
1082	is termed a "group".
1083	b) Sending of the messages where the combination has no semantic significance,
1084	but is merely a convenience or optimisation. This is termed "bundling" – the
1085	combination is termed a "bundle".
1086	
1087	The form A&B is used to refer to a combination (group) where message B is sent in relation
1088	to A ("relation" is asymmetric). The form A+B is used to refer to A and B bundled together-

- the transmission of the bundle "A+B" is semantically identical to the transmission of A
 followed by the transmission of B.
- 1091

1105

- 1092Only certain combinations of messages are possible in a group, and the meaning of the1093relation is specifically defined for each such combination in the next section. A particular1094group is treated as a unit for transmission it has a single target address. This is usually that1095of one of the messages in the group the specification for the group defines which.
- 1097 A "bundle" of messages may contain both unrelated messages and groups of related 1098 messages. The only constraint on which messages and groups can be bundled is that all have the same binding address, but may have different "additional information" values. (Messages 1099 1100 within a related group may have different addresses, where the rules of their relatedness 1101 permit this). Unless constrained by the binding, any messages or groups that are to be sent to the same binding address may be bundled – the fact that the binding addresses are the same is 1102 1103 a necessary and sufficient condition for the sender to determine that the messages can be 1104 bundled.
- 1106A particular and important case of related messages is where a BTP CONTEXT message is1107sent related to an application message. In this case, the target of the application message1108defines the destination of the CONTEXT message. The receiving implementation may in fact1109remove the CONTEXT before delivering the application message to the application (Service)1110proper, but from the perspective of the sender, the two are sent to the same place.1111The compounding mechanisms, and the multi-part address structures, support the "one-wire"1112and "one-shot" communication patterns.
- In "one-wire", all message exchanges between two sides of a Superior: Inferior relationship, 1114 including the associated application messages, pass via the same "endpoints". These 1115 "endpoints" may in fact be relays, routing messages on to particular actors within their 1116 domain. The onward routing will require some further addressing, but this has to be opaque to 1117 the sender. This can be achieved if the relaying endpoint ensures that all addresses for actors 1118 1119 in its domain have the relay's address as their binding address, and any routing information it will need in its own domain is placed in the additional information. (This may involve the 1120 relay changing addresses in messages as they pass through it on the way out). On receiving a 1121 1122 message, it determines the within-domain destination from the received additional information (which is thus rewritten) and forwards the message appropriately. The sender is 1123 1124 unaware of this, and merely sees addresses with the same binding address, which it is permitted to bundle. The content of the "additional information" is a matter only for the relay 1125 - it could put an entire BTP address in there, or other implementation-defined information. 1126 1127 Note that a quite different one-wire implementation can be constructed where there is no 1128 relaying, but the receiving entity effectively performs all roles, using the received identifiers 1129 to locate the appropriate state.
- 1130
- "One-shot" communication makes it possible to send an application message, receive the
 application reply, enrol an Inferior to be responsible for the confirm/cancel of the operations
 of those message and inform the Superior that the Inferior is prepared, all in one two-way
 exchange across the network (e.g. one request/reply of a carrier protocol).. The application
 request is sent with a related CONTEXT message. The application response is sent with a

1136 1137 1138 1139 1140 1141 1142 1143	relation group of CONTEXT_REPLY/related, ENROL/no-rsp-req message and a PREPARED message. This is possible even if the Superior address is different from the address of the application element that sends the original message (if the application exchange is request/reply, there may not even be an identifiable address for the application element). The target addresses of the ENROL and PREPARED (the Superior address) are not transmitted; the actor that was originally responsible for adding the CONTEXT to the outbound application message remembers the Superior address and forwards the ENROL and PREPARED appropriately.
1144 1145 1146 1147 1148 1149	With "one-shot", if there are multiple Inferiors created as a result of a single application message, there is an ENROL and PREPARED message for each sent related to the CONTEXT_REPLY. If an operation fails, a CANCELLED message is sent instead of a PREPARED.
1150 1151 1152 1153 1154 1155	If the CONTEXT has "superior-type" of "atom", then subsequent messages to the same Service, with the same related CONTEXT/atom, can have their associated operations put under the control of the same Inferior, and only a CONTEXT_REPLY/completed is sent back with the response (if the new operations fail, it will be necessary to send back CONTEXT_REPLY/repudiated, or send CANCELLED). If the "superior type" on the CONTEXT is "cohesive", each operation will require separate enrolment.
1156 1157 1158 1159 1160	Whether the "one-shot" mechanism is used is determined by the implementation on the responding (Inferior) side. This may be subject to configuration and may also be constrained by the application or by the binding in use.
1161	Extensibility
1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172	To simplify interoperation between implementations of this edition of BTP with implementations of future editions, the "must-be-understood" sub-parameter as specified for Qualifiers may be defined for use with any parameter added to an existing message in a future revision of this specification. The default for "must-be-understood" shall be "true", so an implementation receiving an unrecognised parameter without a "false" value for "must-be-understood" shall not accept it (the FAULT value "UnrecognisedParameter" is available, but other errors, including lower-layer parsing/unmarshalling errors may be reported instead). If "must-be-understood" with the value "false" is present as a sub-parameter of a parameter in any message, a receiving implementation should ignore the parameter.
1173 1174 1175	How the sub-parameter is associated with the new parameter is determined by the particular binding.
1176 1177	No special mechanism is provided to allow for the introduction of completely new messages.
1178 1179	Inferior handle
1179 1180 1181	Some of the messages exchanged between a Terminator and a Decider are concerned with the individual Inferiors enrolled with the Decider, and not with the business transaction as a

1100	
1182	whole. These messages distinguish the Inferiors of Decider using an "inferior handle". This is
1183	created by the Decider and is unambiguous within the scope of the Decider.
1184	
1185	The "inferior handle" is distinct from the "inferior identifier" passed on an ENROL message
1186	(among other places). The latter is created by the Inferior (or its enroller) and is required to be
1187	unambiguous within the scope of the address as inferior on the ENROL (and unambiguous
1188	within any of the individual addresses in that set of BTP addresses - the identifier must
1189	identify the Inferior across all the places it might migrate to or that have recovery
1190	responsibility for it).
1191	
1192	The "inferior handle" is only used by the Terminator to refer to the inferiors of the Decider.
1193	In messages between the Decider and its Inferiors, the address-as inferior and inferior
1194	identifier are used.
1195	
1196	Messages
	Wiessages
1197	Qualifiara
1198	Qualifiers
1199	
1200	All messages have a Qualifiers parameter which contains zero or more Qualifier values. A
1201	Qualifier has sub-parameters:
1202	
	Sub-parameter Type

Sup-parameter	туре
qualifier name	string
qualifier group	URI
must-be-understood	Boolean
to-be-propagated	Boolean
content	Arbitrary – depends on type

1203	
1204	Qualifier group ensures the Qualifier name is unambiguous. Qualifiers in the
1205	same group need not have any functional relationship. The qualifier group will
1206	typically be used to identify the specification that defines the qualifier's meaning
1207	and use. Qualifiers may be defined in this or other standard specifications, in
1208	specifications of a particular community of users or of implementations or by
1209	bilateral agreement.
1210	
1211	Qualifier name this identifies the meaning and use of the Qualifier, using a name
1212	that is unambiguous within the scope of the Qualifier group.
1213	
1214	Must-be-understood if this has the value "true" and the receiving entity does
1215	not recognise the Qualifier type (or does not implement the necessary
1216	functionality), a FAULT "UnsupportedQualifier" shall be returned and the
1217	message shall not be processed. Default is "true".
1218	

1219	To-be-propagated if this has the value "true" and the receiving entity passes the
1220	BTP message (which may be a CONTEXT, but can be other messages) onwards
1221	to other entities, the same Qualifier value shall be included. If the value is
1222	"false", the Qualifier shall not be automatically included if the BTP message is
1223	passed onwards. (If the receiving entity does support the qualifier type, it is
1224	possible a propagated message may contain another instance of the same type,
1225	even with the same Content – this is not considered propagation of the original
1226	qualifier.). Default is "false".
1227	
1228	Content the type (which may be structured) and meaning of the content is
1229	defined by the specification of the Qualifier.
1230	
1231	
1232	Messages not restricted to outcome or control relationships.
1233	
1234	The messages in this section are used between various roles.CONTEXT message is used in
1235	the Initiator: Factory relationship (when it is related to BEGIN or to BEGUN), and related to
1236	an application 'message' to propagate the business transaction between parts of the
1237	application.CONTEXT_REPLY is used as the reply to a CONTEXT.REQUEST_STATUS
1238	can be issued to, and STATUS returned by any of Decider, Superior or Inferior. FAULT can
1239	be used on any relationship to indicate an error condition back to the sender of a message.
1240	
1241	CONTEXT
1242	
1243	A CONTEXT is supplied by (or on behalf of) a Superior and related to one or more
1244	application messages. (The means by which this relationship is represented is determined by
1245	the binding and the binding mechanisms of the application protocol.) The "superior type"
1246	parameter identifies whether the Superior will apply the same decision to all Inferiors
1247	enrolled using the same superior identifier ("superior type" is "atom") or whether it may
1248	apply different decisions ("superior type" is "cohesion").
1249	

	Parameter	Туре	
	address-as-superior	Set of BTP addresses	
	superior identifier	Identifier	
	reply-address	BTP address	
	superior type	cohesion/atom	
	qualifiers	List of qualifiers	
1250 1251			
1252	address-as-superior the address	ss to which ENROL and other messages from an	
1253	enrolled Inferior are to be sent. This can be a set of alternative addresses.		
1254			
1255	superior identifier identifies the Superior. This shall be globally unambiguous.		
1256	within the scope of the address as superior		

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

1257				
1258	reply-address the address to	which a replying CONTEXT_REPLY is to be sent.		
1259		ne the CONTEXT is transmitted – it refers to the		
1260	destination of a replying CONTEXT_REPLY for this particular transmission of			
1261	the CONTEXT.			
1262				
1263	superior type identifies whet	her the CONTEXT refers to a Cohesion or an		
1264	Atom. Default is atom.			
1265				
1266				
1267	qualifiers standardised or oth	er qualifiers. The standard qualifier "Transaction		
1268	timelimit" is carried by CONT	· ·		
1269	,			
1270	There is no target address parameter for C	ONTEXT as it is only transmitted in relation to the		
1271	application messages, BEGIN and BEGUI	N.		
1272				
1273	The forms CONTEXT/cohesion and CON	TEXT/atom refer to CONTEXT messages with the		
1274	superior type with the appropriate value.			
1275				
1276				
1277	CONTEXT_REPLY			
1278				
1279		f CONTEXT (related to application message(s)) to		
1280	indicate whether all necessary enrolments have already completed (ENROLLED has been			
1281	received) or will be completed by ENROL messages sent in relation to the			
1282	CONTEXT_REPLY or if an enrolment attempt has failed. CONTEXT_REPLY may be sent			
1283	related to an application message (typically the response to the application message related to			
1284	the CONTEXT). In some bindings the CONTEXT_REPLY may be implicit in the application			
1285	message.			
1286	. .	-		
	Parameter	Туре		
	target-address	BTP address		
	superior address	BTP address		
	superior identifier	Identifier		
	completion_status	complete/related/repudiated		
	Qualifiers	List of qualifiers		
1287				

- 1288target-address the address to which the CONTEXT_REPLY is sent. This shall1289be the "reply-address" from the CONTEXT.12901290
 - Superior-address one of the addresses from the address as superior from the CONTEXT. (The parameter is present in CONTEXT_REPLY to disambiguate the superior identifier.)
 - OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

1292 1293

1295	superior identifier the superior identifier from the CONTEXT		
1296 1297	completion_status: reports whether all enrol operations made necessary by the		
1298 1299	receipt of the earlier CONTEXT message have completed. Values are		
	Value	meaning	

14.40	
completed	All enrolments (if any) have succeeded already
<u><i>R</i></u> related	At least some enrolments are to be performed by ENROL messages related to the CONTEXT_REPLY. All other enrolments (if any) have succeeded already.
repudiated	At least one enrolment has failed. The implications of receiving the CONTEXT have not been honoured.

1301

qualifiers standardised or other qualifiers.

- 1302
- 1302 The form CONTEXT_REPLY/completed, CONTEXT_REPLY/related and
- 1304 CONTEXT_REPLY/repudiated refer to CONTEXT_REPLY messages with status having the
- 1305 appropriate value. The form CONTEXT_REPLY/ok refers to either of
- 1306 CONTEXT_REPLY/completed or CONTEXT_REPLY/related.
- 13071308 If there are no necessary enrolments (e.g. the application messages related to the received
- 1309 CONTEXT did not require the enrolment of any Inferiors), then
- 1310 CONTEXT_REPLY/completed is used. 1311

1312If a CONTEXT_REPLY/repudiated is received, the receiving implementation **must** ensure1313that the business transaction will not be confirmed.

1314

1315 1216 **DECUEST S**

1316 **REQUEST_STATUS** 1317

1318 Sent to an Inferior, Superior or to a Decider to ask it to reply with STATUS. The receiver
1319 may reject the request with a FAULT(StatusRefused).
1320

	Parameter	Туре	
	target address	BTP address	
	reply address	BTP address	
	target-identifier	Identifier	
	Qualifiers	List of qualifiers	
1321			
1322	target address the address to which the REQUEST_STATUS message is sent.		
1323	This can be any of address-as-decider, address-as-inferior or address-as-superior.		
1324			
1325	reply address the address to which the replying STATUS should be sent.		

1326		
1327	target identifier Th	e identifier for the business transaction, or part of business
1328	transaction whose st	atus is sought. If the target-adddres is an address-as-decider,
1329	this parameter shall	be the "transaction-identifier" on the BEGUN message. If the
1330	target-address is an	address-as-inferior, this parameter shall be the "inferior-
1331	identifier" on the EN	NROL message. If the target-address is a an address-as-
1332	superior, this param	eter shall be the "superior-identifier" on the CONTEXT.
1333		
1334	qualifiers standard	sed or other qualifiers.
1335		
1336	Types of FAULT possible (sent	to reply address)
1337		
1338	General	
1339	StatusRefu	sed – if the receiver is not prepared to report its status to the
1340	sender of this messa	age
1341	UnknownT	ransaction – if the target-identifier is unknown
1342		C C
1343		
1344	STATUS	
1345		
1346	Sent by a Inferior, Superior or D	ecider in reply to a REQUEST_STATUS, reporting the
1347	overall state of the transaction tr	ee node represented by the sender.
1348		
	Parameter	Туре
	target address	addross

	Falametei	Туре	
	target address	BTP address	
	respondersaddress	BTP address	
	responders-identifier	Identifier	·
	status	See below	
	qualifiers	List of qualifiers	
1349			
1350	target address the addres	ss to which the STATUS is sent. This will be the reply	
1351	address on the REQUEST		
1352		- 0	
1353	responders-address-the	address of the sender of the STATUS message one of	
1354	address as inferior, address as decider, address as superior(with the responders-		
1355	identifier, this determines who the message is from) If the sender has different		
1356	addresses as multiple roles (as Decider, Inferior or Superior), this shall be the		
1357		UEST_STATUS was received.	
1358			
1359	responders-identifier th	e identifier of the state, identical to the "target-	
1360	identifier" on the REQUE	ST_STATUS.aligned with the responders address. If	
1361		les in the transaction (as Decider, Inferior or Superior),	
1362	this shall be the target ide	ntifier on the REQUEST_STATUS	

1363statusstates the current status of the transaction tree node represented by the1364sender. Some of the values are only issued if the sender is an Inferior. If the1365transaction tree node is both Superior and Inferior (i.e. is a sub-coordinator or1366sub-composer), and two status values would be valid for the current state, it is the1367sender's option which one is used.1368

status value	Meaning from Superior	Meaning from Inferior
Created	Not applicable	The Inferior exists (and is addressable) but it has not been enrolled with a Superior
Enrolling	Not applicable	ENROL has been sent, but ENROLLED is awaited
Active	New enrolment of inferiors is possible	The Inferior is enrolled
Resigning	Not applicable	RESIGN has been sent; RESIGNED is awaited
Resigned	Not applicable	RESIGNED has been received
Preparing	Not applicable	PREPARE has been received; PREPARED has not been sent
Prepared	Not applicable	PREPARED has been sent; no outcome has been received or autonomous decision made
Confirming	Confirm decision has been made or CONFIRM has been received as Inferior but responses from inferiors are pending	CONFIRM has been received; CONFIRMED/response has not bee sent
Confirmed	CONFIRMED/responses have been received from all Inferiors	CONFIRMED/response has been sent
Cancelling	Cancel decision has been made but responses from inferiors are pending	CANCEL has been received or auto-cancel has been decided
Cancelled	CANCELLED has been received from all Inferiors	CANCELLED has been sent
cancel- contradiction	Not applicable	Autonomous cancel decision was made, CONFIRM received; CONTRADICTION has not been received
confirm- contradiction	Not applicable	Autonomous confirm decision was made, CANCEL received; CONTRADICTION has not been received

		status value	Meaning from Superior	r	Meaning from Inferior	
		Hazard	A hazard has been repo at least one Inferior	rted from	A hazard has been discovered; CONTRADICTION has not been received	
		Contradicted	Not applicable		CONTRADICTION has been received	
		Unknown	No state information for target-identifier exists	the	No state information for the target-identifier exists	
		Inaccessible	There may be state infor for this target-identifier b cannot be reached/existe cannot be determined	ut it	There may be state information for this target-identifier but it cannot be reached/existence cannot be determined	
1369 1370	qualifiers standardised or other qualifiers.					
1371 1372	Types of FAULT possible					
1373 1374 1275			General			
1375 1376 1377	FAULT					
1377 1378 Sent in reply t 1379		in reply to var	to various messages to report an error condition			
1077	Parameter		Туре			
		target a	ddress	BTP address Identifier		
		superior	ridentifier			
	inferior	identifier	Identifie	r		
		fault typ	е	See bel	ow	
		fault dat	ta	See below		
		qualifier	S	List of qualifiers		
1380 1381		target	addroce the address to			
1381	target address the address to which the FAULT is sent. This may be the reply address from a received message or the address of the opposite side					
1383	(superior/inferior) as given in a CONTEXT or ENROL message					
1384 1385	superior identifier the superior identifier as on the CONTEXT message and as			er as on the CONTEXT message and as		
1386		used on the ENROL message (present only if the FAULT is sent to the superior).				
1387						
1388 1389		inferior identifier the inferior identifier as on the ENROL message (present only if the FAULT is sent to the inferior)				
1390				- /		

1391	fault type identifies the nature of the error, as specified for each of the main
1392	messages.
1393	
1394	fault data information relevant to the particular error. Each fault type defines the
1395	content of the fault data:
1396	

fault type	meaning	fault data
CommunicationFailure	Any fault arising from the carrier mechanism and communication infrastructure.	Determined by the carrier mechanism and binding specification
DuplicateInferior	An inferior with the same address and identifier is already enrolled with this Superior	The identifier
General	Any otherwise unspecified problem	Free text explanation
InvalidDecider	The address the message was sent to is not valid (at all or for this Terminator and transaction identifier)	The address
InvalidInferior	The Superior is known but the Inferior identified by the address- as-inferior and identifier are not enrolled in it	The Inferior Identity (address-as- inferior and identifier)
InvalidSuperior	The received identifier is not known or does not identify a known Superior	The identifier
StatusRefused	The receiver will not report the request status (or inferior statuses) to this StatusRequestor	Free text explanation
InvalidTerminator	The address the message was sent to is not valid (at all or for this Decider and transaction identifier)	The address
UnknownParameter	A BTP message has been received with an unrecognised parameter	Free text explanation
UnknownTransaction	The transaction-identifier is unknown	The transaction-identifier
UnsupportedQualifier	A qualifier has been received that is not recognised and on which "must-be-Understood" is "true".	Qualifier group and name
WrongState	The message has arrived when the recipient is in an invalid state.	

1399 1400 1401 1402		<i>UnknownParameter</i> q u	A BTP message has been received with an unrecognised parameter	Free text explanation
1402 1403 1404		u Qualifiers standardise	ed or other qualifiers.	
1405 1406			anism used for the transmission of essages in a different order than th	
1407		the "WrongState" FAULT	Γ is not sent and should be ignored	if received.
1408 1409 1410	REQUE	ST_INFERIOR_STATUSE	S, INFERIOR_STATUSES	
1410 1411 1412 1413 1414 1415 1416 1417 1418	any Dec Inferior REQUE just issu other m	cider, Superior or Inferior, a s (if any). Since Deciders a EST_INFERIOR_STATUS the FAULT(StatusRefused),	SES with INFERIOR_STATUSES and INFERIOR_STATUSES is a Decider, these messages are desc	its relationships with but non-Deciders may lso used as a reply to
1419	Message	es used in the outcome	relationships	
1420 1421 1422 1423 1424 1425	CONTE			ed after receipt of a
1426				
		Parameter	type	
		target address	BTP address	
		superior identifier	Identifier	
		reply requested	Boolean	
		reply address	BTP address	
		address-as-inferior	Set of BTP addresses	
		inferior identifier	Identifier	
		Qualifiers	List of qualifiers	
1427			·	
1428 1429 1430		-	ddress to which the ENROL is sent om the CONTEXT message.	t. This will be the

1431	superior identifier.	The superior identifier as on the CONTEXT message
1432		
1433	reply requested tr	ue if an ENROLLED response is required, false otherwise.
1434	Default is false.	
1435		
1436	reply address the	address to which a replying ENROLLED is to be sent, if
1437		true. If this field is absent and "reply requested" is true, the
1438		d be sent to the "address-as-inferior" (or one of them, at
1439	sender's option)	
1440	,	
1441	address-as-inferio	the address to which PREPARE, CONFIRM, CANCEL and
1442		E messages for this Inferior are to be sent.
1443	Set ERIOR_STAT	E messages for this interfor are to be sent.
1443	inforior identifier	n identifier that manchimum by identifies this Infanian This
		an identifier that unambiguously-identifies this Inferior. This
1445	snan be globally un set of BTP-addresse	ambiguous. within the scope of any of the address as inferior
1446	Set of BTF-addresse	步.
1447	auglificare (1 1	
1448	•	ised or other qualifiers. The standard qualifier "Inferior
1449	name" may be prese	ent.
1450		
1451	Types of FAULT possible (sent	to Reply address)
1452	Comoral	
1453	General	
1454	•	<i>erior</i> – if superior identifier is unknown
1455	•	nferior – if inferior with at least one of the set address-as-
1456		same and the same inferior identifier is already enrolled
1457		e – if it is too late to enrol new Inferiors (generally if the
1458	Superior ha	s already sent a PREPARED message to its superior or
1459	terminator,	or if it has already issued CONFIRM to other Inferiors).
1460		
1461		s to an ENROL message with "reply requested" having the
1462		q refers to an ENROL message with "reply requested" having
1463	the value "false"	
1464		
1465		sent in relation to CONTEXT_REPLY/related. ENROL/rsp-
1466	1 51 5	T_REPLY/completed will be used (after the ENROLLED
1467	message has been received.)	
1468		
1469	ENROLLED	
1470		
1471	Sent from Superior in reply to a	n ENROL/rsp-req message, to indicate the Inferior has been
1472	A A A	herefore be included in the termination exchanges)
1473		
	Parameter	Туре

target address

BTP address

		Parameter	Туре	
		inferior identifier	Identifier	
		inferior-handle	Handle	
		Qualifiers	List of qualifiers	1
1474				
1475		target address the address to w	which the ENROLLED is sent. This will be the	
1476			message (or one of the address-as-inferiors if the	
1477		reply address was empty)		
1478 1479		inforior identifior. The inferior i	dentifier as on the ENROL message	
1479			denumer as on the ENKOL message	
1480		inferior handle, the inferior han	dle that will identify this newly enrolled Inferior	I
1482			messages between the Superior (acting as a	
1483			s parameter is optional. The value shall be	
1484		different for each enrolled Inferi	or of the Superior.	
1485				
1486		qualifiers standardised or other	qualifiers.	
1487 1488	Νο ΕΔΙΠ Τ	r messages are issued on receiving		
1489	NOTICEI	i messages are issued on receiving		
1490				
1491	RESIGN			
1492				
1493		-	r to remove the Inferior from the enrolment. This	
1494 1495	by the Infer		ness transaction have had no effect as perceived	
1496	by the inter	nor.		
1497	RESIGN m	nay be sent at any time prior to the	e sending of a PREPARED or CANCELLED	
1498		e (which cannot then be sent). RESIGN may be sent in response to a PREPARE		
1499	message.			
1500		D		
		Parameter	type	
		target address	BTP address	
		superior identifier	identifier	1
		address as inferior	Set of BTP addresses	
		inferior identifier	identifier	
		response requested	Boolean	
		Qualifiers	List of qualifiers	
1501				
1502		•	which the RESIGN is sent. This will be the	
1503 1504		superior address as used on the I	ENROL message.	

1505	superior-identifier Th	e superior identifier as on the ENROL message		
1506				
1507	address-as-inferior T	address-as-inferior-The address as inferior as on the earlier ENROL message		
1508	(with the inferior identi	ifier, this determines who the message is from)		
1509				
1510	inferior-identifier The	inferior identifier as on the earlier ENROL message		
1511				
1512	response-requested i	is set to "true" if a RESIGNED response is required.		
1513				
1514	qualifiers standardised	l or other qualifiers.		
1515	·			
1516	Note RESIGN is equivalent to	readonly vote in some other protocols, but can be issued		
1517	early.			
1518				
1519	Types of FAULT possible (sent to a	address-as-inferior)		
1520				
1521	General			
1522	InvalidSuperio	or – if superior identifier is unknown		
1523	InvalidInferior	- if no ENROL had been received for this address-as-		
1524	inferior and ide	entifier (Inferior Identity)		
1525	WrongState –	if a PREPARED or CANCELLED has already been		
1526	received by the	e Superior from this Inferior		
1527		-		
1528	The form RESIGN/rsp-req refers to	an RESIGN message with "reply requested" having the		
1529	value "true"; RESIGN /no-rsp-req r	refers to an RESIGN message with "reply requested"		
1530	having the value "false"			
1531				
1532				
1533	RESIGNED			
1534				
1535	Sent in reply to a RESIGN/rsp-req	message.		
1536				
	Parameter	Туре		
	target address	BTP address		
	inferior identifier	Identifier		
	qualifiers	List of qualifiers		
1537				
1538	target address the add	dress to which the RESIGNED is sent. This will be the		
1539	•	n the ENROL message.		
1540		-		
1541	inferior identifier The inferior identifier as on the earlier ENROL message for			
1542	this Inferior.			
1543				
1544	qualifiers standardised	l or other qualifiers.		
	-	-		

15451546After receiving this message the Inferior will not receive any more messages with this

- 1547 address-as-inferior and identifier.1548
- 1549 No FAULT messages are issued on receiving RESIGNED.

1550 1551 **PREPARE**

- 1552
- 1553Sent from Superior to an Inferior from whom ENROL but neither CANCELLED nor1554RESIGN have been received, requesting a PREPARED message. PREPARE can be sent after1555receiving a PREPARED message.
- 1556 1557

		Parameter	Туре
		target address	BTP address
		inferior identifier	Identifier
		qualifiers	List of qualifiers
1558			
1559		target address the address to w	which the PREPARE message is sent. When sent
1560		•	ill be the address-as-inferior from the ENROL
1561		message.	
1562		C	
1563		inferior identifier When sent from	om Superior to Inferior, the inferior identifier as
1564		on the earlier ENROL message.	
1565		C C	
1566			
1567		qualifiers standardised or other	qualifiers. The standard qualifier "Minimal
1568		inferior timeout" is carried by PI	REPARE.
1569			
1570			
1571	On receiving	g PREPARE, an Inferior should	reply with a PREPARED, CANCELLED or
1572	RESIGN.		
1573			
1574	Types of FA	AULT possible (sent to Superior a	address)
1575			
1576		General	
1577		<i>InvalidInferior</i> – if infer	ior identifier is unknown, or an inferior-handle
1578		on the inferiors-list is unknown	
1579		<i>WrongState</i> – if a CON	FIRM or CANCEL has already been received by
1580		this Inferior.	
1581			
1582			
1583	PREPARED		
1584			

Sent from Inferior to Superior, either unsolicited or in response to PREPARE, but only when
the Inferior has determined the operations associated with the Inferior can be confirmed and
can be cancelled, as may be instructed by the Superior. The level of isolation is a local matter
(i.e. it is the Inferiors choice, as constrained by the shared understanding of the application
exchanges) – other access may be blocked, may see applied results of operations or may see
the original state.

1591

1571	- .	_
	Parameter	Туре
	target address	BTP address
	superior identifier	Identifier
	address as inferior	Set of BTP addresses
	inferior identifier	Identifier
	default is cancel	Boolean
	qualifiers	List of qualifiers
1592		
1593	target address the address to wh	nich the PREPARED is sent. This will be the
1594	Superior address as on the ENRO	
1595	L L	C
1596	superior identifier When the me	ssage is sent from an Inferior to the Superior,
1597	the superior identifier as on the E	NROL message
1598		
1599	address-as-inferior - When the m	essage is sent from an Inferior to the Superior,
1600	the address as inferior as on the e	arlier ENROL message (with the inferior
1601	identifier, this determines who the	e message is from)
1602		
1603	inferior identifier The inferior id	entifier as on the ENROL message
1604		
1605		ferior states that if the outcome at the Superior
1606		ted with this Inferior, no further messages need
1607		ior does not receive a CONFIRM message, it
1608	×	ons. The value "true" will invariably be used
1609		what circumstances (usually a timeout) an
1610		Il be made. If "false", the Inferior will expect
1611		ge as appropriate, even if qualifiers indicate that
1612	an autonomous decision will be n	hade.
1613		
1614		ualifiers. The standard qualifier "Inferior
1615	timeout" may be carried by PREF	ARED.
1616	On conding a DDEDADED, the Inferior under	takas to maintain its shility to confirm or concel
1617 1618		takes to maintain its ability to confirm or cancel
1618	the effects of the associated operations until it Qualifiers may define a time limit or other con	÷
1017	Quantiers may define a time mint of other con	isuants on this promise. The default is

1620 1621	cancel" parameter affects only the subsequent message exchanges and does not of itself state that cancellation will occur.			
1622 1623 1624	Types of FAULT possible (sent to address-as-inferior)			
1624	General			
1625		<i>General</i> InvalidSuperior – if Superior identifier is unknown		
1620		<i>ior</i> – if no ENROL has been received for this address-as-		
1627		identifier, or if RESIGN has been received for this Inferior		
1628	interior and	identifier, of it RESION has been received from this interior		
1630	The form PREPARED/cancel re-	fers to a PREPARED message with "default is cancel" –		
1630		form PREPARED/cancel refers to a PREPARED message with "default is cancel" = 2. The unqualified form PREPARED refers to a PREPARED message with "default is		
1632	cancel" = "false".	LITINED TOTOTS to a TREET TREED THOUSSAGE with actual is		
1633				
1634				
1635	CONFIRM			
1636				
1637	Sent by the Superior to an Inferio	or from whom PREPARED has been received.		
1638				
	Parameter	Туре		
	target address	BTP address		
	inferior identifier	Identifier		
	qualifiers	List of qualifiers		
1639				
1640	•	address to which the CONFIRM message is sent. This will		
1641	be the address-as-inf	be the address-as-inferior from the ENROL message.		
1642				
1643		he inferior identifier as on the earlier ENROL message for		
1644	this Inferior.			
1645		1 (1 1) (*		
1646	quaimers standardi	sed or other qualifiers.		
1647	On reassiving CONFIDM, the Int	ferior is released from its promise to be able to undo the		
1648 1649	U	Inferior. The effects of the operations can be made available		
1650	to everyone (if they weren't alread	A		
1650	to everyone (if they weren t area	uuy).		
1652	Types of FAULT possible (sent	to Superior address)		
1653				
1654	General			
1655		<i>ior</i> – if inferior identifier is unknown		
1656		<i>e</i> – if no PREPARED has been sent by, or if CANCEL has		
1657		ed by this Inferior.		
1658				
1659				

CONFIRMED 1660

1661

Sent after the Inferior has applied the confirmation, both in reply to CONFIRM or when the 1662 Inferior has made an autonomous confirm decision, and in reply to a 1663 CONFIRM_ONE_PHASE if the Inferior decides to confirm its associated operations.

1664

1665 1666

1000		
	Parameter	Туре
	target address	BTP address
	superior identifier	Identifier
	address as inferior	Set of BTP addresses
	inferior identifier	Identifier
	confirm received	Boolean
	qualifiers	List of qualifiers
1667		
1668	target address the address to	which the CONFIRMED is sent. When sent by an
1669		ill be the Superior address as on the CONTEXT
1670	message.	
1671	e	
1672	superior identifier When the	message is sent from an Inferior to the Superior,
1673		ifier as on the CONTEXT message.
1674	L L	C
1675	address-as-inferior When the	e message is sent from an Inferior to the Superior,
1676		Ferior as on the earlier ENROL message (with the
1677	inferior identifier, this determi	ines who the message is from).
1678		
1679	inferior identifier When the r	nessage is sent from an Inferior to the Superior, this
1680	shall be the inferior identifier	as on the earlier ENROL message.
1681		
1682		
1683		
1684		ONFIRMED is sent after receiving a CONFIRM
1685		nous confirm decision has been made and either if
1686		een received or the implementation cannot
1687		een received (due to loss of state information in a
1688	failure).	
1689	11 11	
1690	qualifiers standardised or oth	er qualifiers.
1691		· c ·)
1692	Types of FAULT possible (sent to address	s-as-interior)
1693	General	
1694		Venezioni dentificario eralmorra
1695	invaliuSuperior – if S	Superior identifier is unknown

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

1696 1697 1698			- if no ENROL has been received for this address-as- ntifier, or if RESIGN has been received from this Inferior.	
1699 1700 1701 1702	Note – A CONFIRMED message arriving before a CONFIRM message is sent, or after a CANCEL has been sent will occur when the Inferior has taken an autonomous decision and is not regarded as occurring in the wrong state. (The latter will cause a CONTRADICTION message to be sent.)			
1703 1704 1705 1706 1707 1708			D/auto refers to a CONFIRMED message with "confirm ONFIRMED/response refers to a CONFIRMED message " = "true".	
1709	CANCEL			
1710 1711 1712	Sent by t	the Superior to an Inferior a	t any time before (and unless) CONFIRM has been sent.	
		Parameter	Туре	
		target address	BTP address	
		inferior identifier	Identifier	
		qualifiers	List of qualifiers	
1713		1		
1714		target address the add	ress to which the CANCEL message is sent. When sent	
1715		from Superior to Inferior, tThis will be the address-as-inferior from the ENROL		
1716		message.		
1717 1718		inferior identifier Whe	n sent from Superior to Inferior, the inferior identifier as	
1719		on the earlier ENROL n		
1720				
1721		qualifiers standardised	or other qualifiers.	
1722	XX 71			
1723 1724	When sent to an received by an Inferior, the effects of any operations associated with the Inferior should be undone. If the Inferior had sent PREPARED, the Inferior is released from			
1725	Inferior should be undone. If the Inferior had sent PREPARED, the Inferior is released from its promise to be able to confirm the operations.			
1726	I I			
1727	Types of FAULT possible (sent to Superior address)			
1728		0 1		
1729		General Investigation		
1730 1731		on the inferiors-list is u	- if inferior identifier is unknown, or an inferior-handle	
1731			f a CONFIRM has been received by this Inferior.	
1732		wionystate - 1		
1734				

I

1735 1736	CANCELLED	
1737 1738 1739		(or is applying) cancellation of the operations associated sent from Inferior to Superior in the following cases:
1740 1741 1742		sending PREPARED, to indicate the Inferior is unable to full and is cancelling all of them;
1743 1744 1745	2. in reply to CANCEL, 1	regardless of whether PREPARED has been sent;
1746 1747	3. after sending PREPAR decision to cancel.	ED and then making and applying an autonomous
1748 1749 1750 1751	4. in reply to CONFIRM associated operations	_ONE_PHASE if the Inferior decides to cancel the
1751 1752 1753 1754	As is specified in the state tables, c circumstances of recovery and rese	ases 1, 2 and 3 are not always distinct in some ending of messages.
1751	Parameter	
	target address	BTP address
	superior identifier	Identifier
	address as inferior	Set of BTP address
	inferior identifier	Identifier
	qualifiers	List of qualifiers
1755		
1756 1757	0	dress to which the CANCELLED is sent. When sent by an Tthis will be the Superior address as on the CONTEXT
1758	message.	Tenis will be the Superior address as on the CONTEXT
1759		
1760	•	hen the message is sent from an Inferior to the Superior,
1761	this shall be the superior	or identifier as on the CONTEXT message.
1762	addross as inferior W	Then the masses is sent from on Inferior to the Superior
1763 1764		Then the message is sent from an Inferior to the Superior, as as inferior as on the earlier ENROL message (with the
1765		determines who the message is from).
1766		
1767		en the message is sent from an Inferior to the Superior, this
1768	shall be the inferior ide	entifier as on the earlier ENROL message.
1769 1770	qualifiers standardise	d or other qualifiers
1770	quaimers standardise	d'or other quainners.
1772	Types of FAULT possible (sent to	address-as-inferior)

1773 1774 1775 1776 1777 1778 1779		<i>General</i> <i>InvalidSuperior</i> – if Superior identifier is unknown <i>InvalidInferior</i> – if no ENROL has been received for this address-as- inferior and identifier, or if RESIGN has been received from this Inferior <i>WrongState</i> – if CONFIRM has been sent			
1780			Note – A CANCELLED message arriving before a CANCEL message is		
1781		-	M has been sent will occur when the Inferior has		
1782			cision and is not regarded as occurring in the wrong		
1783		state. (The latter will cau	ise a CONTRADICTION message to be sent.)		
1784					
1785					
1786	CONFIRM_0	ONE_PHASE			
1787 1788	Sent fro	m a Superior to an enroll	ed Inferior, when there is only one such enrolled Inferior. In		
1789			is not performed between the Superior and Inferior and the		
1790			ons associated with the Inferior is determined by the Inferior.		
1791		· · · · · · · · · · · · · · · · · · ·			
		Parameter	Туре		
		target address	BTP address		
		inferior identifier	Identifier		
		report-hazard	boolean		
		qualifiers	List of qualifiers		
1792					
1793		target address the a	address to which the CONFIRM_ONE_PHASE message is		
1794		sent This will be the	address-as-inferior on the ENROL message.		
1795		inforior identifier m			
1796 1797	inferior identifier The inferior identifier as on the earlier ENROL message for this Inferior.				
1797		uns mierior.			
1799		report hazard Defin	es whether the superior wishes to be informed if a mixed		
1800	report hazard Defines whether the superior wishes to be informed if a mixed condition occurs for the operations associated with the Inferior. If "report hazard"				
1801			will reply with HAZARD if a mixed condition occurs, or if		
1802			etermine that a mixed condition has not occurred. If "report		
1803			nferior will report only its own decision, regardless of		
1804		whether that decision	was correctly and consistently applied. Default is false.		
1805		muelifiers to the	1 .1 110		
1806		quaimers standardis	ed or other qualifiers.		
1807					

1808 CONFIRM ONE PHASE can be issued by a Superior to an Inferior from whom PREPARED has been received (subject to the requirement that there is only one enrolled 1809 1810 Inferior). 1811 1812 Types of FAULT possible (sent to Superior address) 1813 1814 General *InvalidInferior* – if inferior identifier is unknown 1815 *WrongState* – if a PREPARE has already been received fromsent to this 1816 Inferior 1817 1818 HAZARD 1819 1820 1821 Sent when the Inferior has either discovered a "mixed" condition: that is unable to correctly 1822 and consistently cancel or confirm the operations in accord with the decision (either the received decision of the superior or its own autonomous decision), or when the Inferior is 1823 unable to determine that a "mixed" condition has not occurred. 1824 1825 1826 HAZARD is also used to reply to a CONFIRM_ONE_PHASE if the Inferior determines there is a mixed condition within its associated operations or is unable to determine that there is not 1827 a mixed condition. 1828 1829

	Parameter	Гуре
	target address	BTP address
	superior identifier	Identifier
	address as inferior	Set of BTP addresses
	inferior identifier	Identifier
	level	mixed/possible
	Qualifiers	List of qualifiers
1830		
1831	target address the address to wh	nich the HAZARD is sent. This will be the
1832	superior address from the ENROL message.	
1833	superior address from the Erticor	2 mossuge.
1834	superior identifier. The superior	identifier as used on the ENROL message
1835	Superior identifier The superior	identifier as used on the EP(KOE message
1836	address as inferior. The address	as inferior as on the earlier ENROL message
1837	(with the inferior identifier, this determines who the message is from)	
1838	(with the interior identifier, this d	ciermines who ure message is from
1839	inferior identifier. The inferior id	entifier as on the earlier ENROL message
1840	interior identifier The interior id	entitier as on the earlier ENROL message
1841	lovel indicated with value "mixed	1" that a mixed condition has definitely
		I" that a mixed condition has definitely
1842	-	" that it is unable to determine whether a mixed
1843	condition has occurred or not.	

Tuno

Daramator

1044			
1844			
1845	qualifiers standardised or other qualifiers.		
1846 1847	Types of FAULT possible (sent to address-as-inferior)		
1847	Types of FAULT possible (sen	to address-as-interior)	
1848	General		
		porior if Sumarian identifian is unly sum	
1850	-	<i>Derior</i> – if Superior identifier is unknown	
1851		rior – if no ENROL has been received for this address-as-	
1852 1853	interior and	l identifier, or if RESIGN has been received from this Inferior	
1855			
1854	The form $HAZARD$ /mixed refe	ers to a HAZARD message with "level" = "mixed", the form	
1855		HAZARD message with "level" = "possible".	
1857		n Ex message whith level - possible .	
1858	CONTRADICTION		
1859			
1860	Sent by the Superior to an Infer	ior that has taken an autonomous decision contrary to the	
1861		etected by the Superior when the 'wrong' one of	
1862		D is received. CONTRADICTION is also sent in response to a	
1863	HAZARD message.	*	
1864			
	Parameter	Туре	
	target address	BTP address	
	inferior identifier	Identifier	
	Qualifiers	List of qualifiers	
1865			
1866	•	e address to which the CONTRADICTION message is sent.	
1867	This will be the address-as-inferior from the ENROL message.		
1868			
1869	inferior identifier The inferior identifier as on the earlier ENROL message for		
1870	this Inferior.		
1871			
1872	qualifiers standardised or other qualifiers.		
1873			
1874 1875	Types of FAULT possible (sen	to Superior address)	
1875	General		
		rior if informing identifier is unknown	
1877 1878		rior – if inferior identifier is unknown	
1878 1879	by this Infe	<i>te</i> – if neither CONFIRMED or CANCELLED has been sent	
1879	by this life	4101	
1880	SUPERIOR_STATE		
1881			
1883	Sent by a Superior as a query to	an Inferior when	
1884	zoni og a superior as a query te		

1885 1886	1.	in the active state	
1880 1887 1888 1889	2. there is uncertainty what state the Inferior has reached (due to recovery from previous failure or other reason).		
1889 1890 1891 1892	Also sent by the Superior to the Inferior in response to a received INFERIOR_STATE, in particular states.		
1072		Parameter	Туре
		target address	BTP address
		inferior identifier	Identifier
		Status	see below
		reply requested	Boolean
		Qualifiers	List of qualifiers
1893 1894 1895 1896 1897 1898 1899 1900		 target address the address to which the SUPERIOR_STATE message is sent. This will be the address-as-inferior from the ENROL message. inferior identifier The inferior identifier as on the earlier ENROL message for this Inferior. status states the current state of the Superior, in terms of its relation to this 	
1900 1901 1902		Inferior only.	it state of the Superior, in terms of its relation to this
		status value	Meaning
		active	The relationship with the Inferior is in the active state from the perspective of the Superior; ENROLLED has been sent, PREPARE has not been sent and PREPARED has not been received (as far as the Superior knows)
		prepared-received	PREPARED has been received from the Inferior, but no outcome is yet available
		inaccessible	The state information for the Superior, or for its relationship with this Inferior, if it exists, cannot be accessed at the moment. This should be a transient condition
		unknown	The Inferior is not known – it does not exist from the perspective of the Superior. The Inferior can treat this as an instruction to cancel any associated operations
1903		Dank, some stad	
1904 1905			if SUPERIOR_STATE is sent as a query at the Superior's ERIOR_STATE is sent in reply to a received
1905			other message. Can only be true if status is active or
1007		prepared_received	

1907 prepared-received.

1908			
1909	qualifiers standardised	l or other qualifiers.	
1910			
1911	The Inferior, on receiving SUPERIC	DR_STATE with reply requested = true, should reply in a	
1912	timely manner by (depending on its	state) repeating the previous message it sent or by	
1913	sending INFERIOR_STATE with the	he appropriate status value.	
1914			
1915	A status of unknown shall only be s	ent if it has been determined for certain that the Superior	
1916	has no knowledge of the Inferior, or	(equivalently) it can be determined that the relationship	
1917	with the Inferior was cancelled. If the	nere could be persistent information corresponding to the	
1918	Superior, but it is not accessible from	m the entity receiving an INFERIOR_STATE/*/y (or	
1919	other) message targeted to the Supe	rior or that entity cannot determine whether any such	
1920	persistent information exists or not,	the response shall be Inaccessible.	
1921			
1922	SUPERIOR_STATE/unknown is al	so used as a response to messages, other than	
1923	INFERIOR_STATE/*/y that are rec	ceived when the Inferior is not known (and it is known	
1924	there is no state information for it).		
1925			
1926	The form SUPERIOR_STATE/abc	d refers to a SUPERIOR_STATE message status having a	
1927	value equivalent to "abcd" (for active, prepared-received, unknown and inaccessible) and		
1928	with "reply requested" = "false". SU	JPERIOR_STATE/abcd/y refers to a similar message, but	
1929	with "reply requested" = "true". The	e form SUPERIOR_STATE/*/y refers to a	
1930	SUPERIOR_STATE message with	"reply requested" = "true" and any value for status.	
1931			
1932			
1933	INFERIOR_STATE		
1934			
1935	Sent by an Inferior as a query when in the active state to a Superior, when (due recovery from		
1936	previous failure or other reason) there is uncertainty what state the Superior has reached.		
1937			
1938	Also sent by the Inferior to the Superior in response to a received SUPERIOR_STATE, in		
1939	particular states.		
1940			
	Parameter	Туре	
	target address	BTP address	
	superior identifier	Identifier	

target address	BTP address
superior identifier	Identifier
address as inferior	BTP address
inferior identifier	Identifier
Status	see below
reply requested	Boolean
Qualifiers	List of qualifiers

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

1942	target address the a	target address the address to which the INFERIOR_STATE is sent. This will		
1943	be the target address	be the target address as used the original ENROL message.		
1944	-			
1945	superior identifier	superior identifier The superior identifier as used on the ENROL message		
1946	•			
1947	address-as-inferior-	The address as inferior as on the ENROL message (with the		
1948		s determines who the message is from)		
1949	interior identifier, un	s determines who the message is nomy		
1950	inforior identifier T	ne inferior identifier as on the ENROL message		
		le interior identifier as on the ENROL message		
1951				
1952		rent state of the Inferior for the atomic business transaction,		
1953		the last message sent to the Superior by (or in the case of		
1954	ENROL for) the Infe	rior		
1955				
	status value	meaning/previous message sent		
	active	The relationship with the Superior is in the active state from the perspective of the Inferior; ENROL has been sent, a decision to send PREPARED has not been made.		
	inaccessible	The state information for the relationship with the Superior, if it exists, cannot be accessed at the moment. This should be a transient condition		
	unknown	The Inferior is not known – it does not exist from the perspective of the Superior. The Inferior can be treated as cancelled		
1956		·		
1957	reply requested "tra	ue" if INFERIOR_STATE is sent as a query at the		
1958		"false" if INFERIOR_STATE is sent in reply to a received		
1959				
1960		SUPERIOR_STATE or other message. Can only be "true" if "status" is "active" or "propagad received". Can only be "true" if "status" is "active"		
1961	or prepared-received	or "prepared-received". Can only be "true" if "status" is "active".		
1962	qualifiers standardis	and on other qualifiers		
	quaimers standardis	ed or other qualifiers.		
1963	The Superior on receiving NEE	DIOD STATE with "marked an analysis of all second and the second se		
1964		RIOR_STATE with "reply requested" = "true", should reply		
1965		g on its state) repeating the previous message it sent or by		
1966	sending SUPERIOR_STATE wit	n the appropriate status value.		
1967				
1968		be sent if it has been determined for certain that the Inferior		
1969	has no knowledge of a relationship with the Superior. If there could be persistent information			
1970		t it is not accessible from the entity receiving an		
1971		r) message targetted on the Inferior or the entity cannot		
1972	• •	istent information exists, the response shall be		
1973	"inaccessible".			
1974				
1975		also used as a response to messages, other than		
1976	SUPERIOR_STATE/*/y that are	received when the Inferior is not known (and it is known		
1977	there is no state information for it	t).		

1978	
1979	A SUPERIOR_STATE/INFERIOR_STATE exchange that determines that one or both sides
1980	are in the active state does not require that the Inferior be cancelled (unlike some other two-
1981	phase commit protocols). The relationship between Superior and Inferior, and related
1982	application elements may be continued, with new application messages carrying the same
1983	CONTEXT. Similarly, if the Inferior is prepared but the Superior is active, there is no
1984	required impact on the progression of the relationship between them.
1985	
1986	The form INFERIOR_STATE/abcd refers to a INFERIOR_STATE message status having a
1987	value equivalent to "abcd" (for active, unknown and inaccessible) and with "reply requested"
1988	= "false". INFERIOR_STATE/abcd/y refers to a similar message, but with "reply requested"
1989	= "true". The form INFERIOR_STATE/*/y refers to a INFERIOR_STATE message with
1990	"reply requested" = "true" and any value for status.
1991	
1992	
1993	
1994	
1995	REDIRECT
1996	
1997	Sent when the address previously given for a Superior or Inferior is no longer valid and the

1997 Sent when the address previously given for a Superior or Inferior is no longer valid and the
1998 relevant state information is now accessible with a different address (but the same superior or
1999 inferior identifier).
2000

	Parameter	Туре
	target address	BTP address
	superior identifier	Identifier
	inferior identifier	Identifier
	old address	Set of BTP addresses
	new address	Set of BTP addresses
	qualifiers	List of qualifiers
2001		
2002	0	which the REDIRECT is sent. This may be the
2003	reply address from a received message or the address of the opposite side	
2004	(superior/inferior) as given in a CONTEXT or ENROL message	
2005		
2006	•	r identifier as on the CONTEXT message and
2007	used on an ENROL message. (present only if the REDIRECT is sent from the	
2008	Inferior).	
2009		
2010	inferior identifier The inferior	identifier as on the ENROL message
2011		
2012	old address The previous addr	ess of the sender of REDIRECT. A match is
2013	considered to apply if any of the	e old addresses match one that is already known.
2014		

2015	new address T	new address The (set of alternatives) new addresses to be used for messages	
2016	sent to this entity	<i>.</i>	
2017			
2018	qualifiers stand	qualifiers standardised or other qualifiers.	
2019			
2020		e address is changed is an Inferior, the new address value	
2021	replaces the add	replaces the address-as-inferior as present in the ENROL.	
2022			
2023		e address is changed is a Superior, the new address value	
2024		erior address as present in the CONTEXT message (or as present	
2025 2026	in any other med	hanism used to establish the Superior:Inferior relationship).	
2020			
2027			
2029	Massagas used in control	valationshing	
2029	Messages used in control	erationships	
2030	BEGIN		
2031	DEGIN		
2032	A request to a Factory to cre	ate a new Business Transaction. This may either be a new top-	
2034	· · ·	se the Composer or Coordinator will be the Decider, or the new	
2035	Business Transaction may be immediately made the Inferior within an existing Business		
2036	•	sub-Composer or sub-Coordinator).	
2037	-	-	
	Parameter	Туре	
	target address	BTP address	
	reply address	BTP address	
	transaction type	cohesion/atom	
	qualifiers	List of qualifiers	
2038			
2039	target address	the address of the entity to which the BEGIN is sent. How this	
2040	address is acquir	ed and the nature of the entity are outside the scope of this	
2041	specification.		
2042			
2043	1 5	he address to which the replying BEGUN and related	
2044	CONTEXT mes	sage should be sent.	
2045			
2046		e identifies whether a new Cohesion or new Atom is to be	
2047	created; this value	e will be the "superior type" in the new CONTEXT	
2048			

2049qualifiers standardised or other qualifiers. The standard qualifier "Transaction2050timelimit" may be present on BEGIN, to set the timelimit for the new business2051transaction and will be copied to the new CONTEXT. The standard qualifier2052"Inferior name" may be present if there is a CONTEXT related to the BEGIN.2053

2054	A new top-level Business Transaction is created if there is no CONTEXT related to the			
2055	BEGIN. A Business Transaction that is to be Inferior in an existing Business Transaction is			
2056	created if the CONTEXT message for the existing Business Transaction is related to the			
2057	BEGIN. In this case, the Factory is responsible for enrolling the new Composer or			
2058	Coordinator as an Inferior of the Superior identified in that CONTEXT.			
2059				
2060	-	es not provide a standardised means to		
2061		ors of a sub-Composer are in its confirm set.		
2062	This is considered part of the	application:inferior relationship.		
2063				
2064	The forms BEGIN/cohesion and BEG	IN/atom refer to BEGIN with "transaction type" having		
2065	the corresponding value.			
2066		1 11 \		
2067 2068	Types of FAULT possible (sent to Re	ply address)		
2069	General			
2070				
2071	BEGUN			
2072 2073	BEGUN is a reply to BEGIN. There i	s always a related CONTEXT, which is the CONTEXT		
2074	for the new business transaction.			
2075				
	Parameter	Туре		
	target address	BTP address		
	address-as-decider	Set of BTP addresses		
	transaction-identifier	Identifier		
	inferior-handle	Handle		
	address-as-inferior	Set of BTP addresses		
	qualifiers	List of qualifiers		
2076				
2077	U	ess to which the BEGUN is sent. This will be the reply		
2078	address from the BEGIN.	address from the BEGIN.		
2079	address as desider for			
2080 2081		address-as-decider for a top-level transaction (no CONTEXT related to the		
2081		BEGIN), this is the address to which PREPARE_INFERIORS, CONFIRM_TRANSACTION, CANCEL_TRANSACTION,		
2082	CANCEL_INFERIORS and REQUEST_INFERIOR_STATUSES messages are			
2084		Γ was related to the BEGIN this parameter is absent		

2086transaction-identifieridentifies the new Decider (Composer or Coordinator)2087within the scope of the address-as-decider. If this is not a top-level transaction,2088the transaction-identifier is optional, but if present shall be the inferior-identifier

2089	used in the enrolment with the Superior identified by the CONTEXT related to
2090	the BEGIN.
2091	
2092	inferior handle Shall be absent if this is a top-level transaction and may or may
2093	not be present otherwise. (Presence or absence will be determined by the nature
2094	of the Superior identified in the CONTEXT related to the BEGIN). If present, the
2095	inferior handle will identify this new business transaction as in the inferiors-list
2096	parameters in messages between the Superior identified in the CONTEXT related
2097	to the BEGIN (acting as a Decider) and its Terminator. The value shall be
2098	different for each enrolled Inferior of that Superior.
2099	
2100	address-as-inferior This parameter shall be absent if this is a top-level
2101	transaction and may be present, at implementation option otherwise. If present, it
2102	shall be the address-as-inferior used in the enrolment with the Superior identified
2103	by the CONTEXT related to the BEGIN. If this is a top-level transaction
2104	
2105	qualifiers standardised or other qualifiers.
2106	
2107	At implementation option, the "address-as-decider" and/or "address-as-inferior" and the
2108	"address-as-superior" in the related CONTEXT may be the same or may be different. There
2109	is no general requirement that they even use the same bindings. Any may also be the same as
2110	the target address of the BEGIN message (the inferior identifier on messages will ensure they
2111	are applied to the appropriate Composer or Coordinator).
2112	
2113	No FAULT messages are issued on receiving BEGUN.
2114	
2115	PREPARE_INFERIORS
2116	
2117	Sent from a Terminator to a Decider, but only if it is a Cohesion Composer, to tell it to
2118	prepare all or some of its inferiors, by sending PREPARE to any that have not already sent
2119	PREPARED, RESIGN or CANCELLED to the Decider (Composer) on its relationships as
2120	Superior. If the inferiors-list parameter is absent, the request applies to all the inferiors; if the
2121	parameter is present, it applies only to the identified inferiors of the Decider (Composer).
2122	

	Parameter	Туре
	target address	BTP address
	reply address	BTP address
	transaction-identifier	Identifier
	inferiors-list	List of Identifiersinferior handles
	qualifiers	List of qualifiers
2123		
2124	target address the address to w	hich the PREPARE_INFERIORS message is
2125	sent. This will be the decider-add	lress from the BEGUN message.
2126		

l

2127	reniv address the address	of the Terminator sending the
2127	PREPARE_INFERIORS m	-
2129		
2129	transaction identifier iden	tifies the Decider and will be the transaction-identifier
2130	from the BEGUN message.	
2131	nom the DECON message.	
2132	inferiors-list defines which	of the Inferiors of this Decider preparation is
2133		ferior-identifiers" as on the ENROL received by the
2135		tior). If this parameter is absent, the PREPARE
2136	applies to all Inferiors.	,
2137		
2138	qualifiers standardised or	other qualifiers.
2139	l l	1
2140		
2141	For all Inferiors identified in the inferio	rs-list parameter (all Inferiors if the parameter is
2142	absent), from which none of PREPARE	D, CANCELLED or RESIGNED has been received,
2143	the Decider shall issue PREPARE. It was	ll reply to the Terminator, using the reply address on
2144		ending an INFERIOR_STATUSES message giving
2145		he inferiors-list parameter (all of them if the
2146	parameter was absent).	
2147		
2148	Types of FAULT possible (sent to Supe	rior address)
2149		
2150	General	
2151		Decider address is unknown
2152		tion – if the transaction-identifier is unknown
2153	<i>InvalidInferior</i> – if	an inferior-handle on the inferiors-list is unknown
2154	<i>WrongState</i> – if a	CONFIRM_TRANSACTION or
2155	CANCEL_TRANS	ACTION has already been received by this
2156	Composer.	
2157		
2158		efers to a PREPARE_INFERIORS message where
2159		The form PREPARE_INFERIORS/specific refers to a
2160 2161	PREPARE_INFERIORS message when	e the "inferiors-list" parameter is present.
2161		I
2162		
2163	CONFIRM_TRANSACTION	
2164		
2165	Sent from a Terminator to a Decider to	request confirmation of the business transaction. If the
2160		confirm-set is specified by the "inferiors-list"
2168	parameter.	
2169	r	
	Parameter	Туре
	target address	BTP address
	larget audiess	ווע מעמולא

	reply address	BTP address	
	transaction identifier	Identifier	
	inferiors-list	List of inferior handles Identifiers	
	report-hazard	Boolean	l
	Qualifiers	List of qualifiers	
2170	Qualmers		
2170	target address the address t	to which the CONFIRM_TRANSACTION message	
2172	0	ess-as-decider on the BEGUN message.	
2173			
2174	reply address the address o		
2175	CONFIRM_TRANSACTIO	N message.	
2176			
2177		fies the Decider. This will be the transaction-	
2178 2179	identifier from the BEGUN r	nessage.	
2179	inferiors-list defines which	Inferiors enrolled with the Decider, if it is a	
2180		e confirmed, using the "inferior-identifiers" as on	
2182		Decider (in its role as Superior). Shall be absent if	
2183	the Decider is an Atom Coor		1
2184			
2185	report hazard Defines whet	her the Terminator wishes to be informed of hazard	
2186		isions within the business transaction. If "report	
2187		r will wait until responses (CONFIRMED,	
2188	-) have been received from all of its inferiors,	
2189		nts are reported. If "report hazard" is "false", the	
2190 2191	soon as the decision for the tr	FIRM_COMPLETE or CANCEL_COMPLETE as	
2191	soon as the decision for the t	ansaction is known.	
2192	qualifiers standardised or ot	her qualifiers	
2193		nor quanters.	
2195	If the "inferiors-list" parameter is present	, the Inferiors identified shall be the "confirm-set" of	
2196		nd the business transaction is a Cohesion, the	
2197		riors. If the business transaction is an Atom, the	
2198	"confirm-set" is automatically all the Infe	eriors.	
2199			
2200	Any Inferiors from which RESIGN is rec	erved are not counted in the confirm-set.	
2201 2202	If for each of the Inferiors in the confirm	-set, PREPARE has not been sent and PREPARED	
2202	has not been received, PREPARE shall b		
2203			
2205	NOTE If PREPARE has been	sent but PREPARED not yet received from	
2206	an Inferior in the confirm-set, it	is an implementation option whether and	
2207	when to re-send PREPARE. The	Superior implementation may choose to re-	

2208 2209	send PREPARE if there are indicated delivered.	ations that the earlier PREPARE was not
2210		
2211		
2212	A confirm decision may be made only if P	REPARED has been received from all Inferiors in
2213	• •	sion shall be persistent (and if it is not possible to
2214		e is only one remaining Inferior in the "confirm
2215	-	, CONFIRM_ONE_PHASE may be sent to it.
2216		
2217	All remaining Inferiors that are not in the o	confirm set shall be cancelled.
2218		
2219	If a confirm decision is made and "report-h	nazard" was "false", a CONFIRM_COMPLETE
2220	message shall be sent to the "reply-address	
2221		
2222	If a cancel decision is made and "report-ha	zard" was "false", a CANCEL_COMPLETE
2223	message shall be sent to the "reply-address	
2224		
2225	If "report-hazard" was "true" and any HAZ	ZARD or contradictory message was received (i.e.
2226	CANCELLED from an Inferior in the cont	firm-set or CONFIRMED from an Inferior not in
2227	the confirm-set), an INFERIOR_STATUS	ES reporting the status for all Inferiors shall be sent
2228	to the "reply-address".	
2229		
2230	Types of FAULT possible (sent to reply ac	ldress)
2231		
2232	General	
2233	<i>InvalidDecider</i> – if De	ecider address is unknown
2234	UnknownTransaction	η – if the transaction-identifier is unknown
2235	<i>InvalidInferior</i> – if an	inferior handle in the inferiors-list is unknown
2236	<i>WrongState</i> – if a CA	NCEL_TRANSACTION has already been
2237	received.	
2238		
2239	The form CONFIRM_TRANSACTION/al	ll refers to a CONFIRM_TRANSACTION message
2240	where the "inferiors-list" parameter is abse	ent. The form
2241	CONFIRM_TRANSACTION/specific ref	ers to a CONFIRM_TRANSACTION message
2242	where the "inferiors-list" parameter is pres	
2243		
2244	TRANSACTION_CONFIRMED	
2245		
2246	A Decider sends TRANSACTION_CONF	
2247		confirm-set confirms (and, for a Cohesion, all other
2248		s, or if the Decider made a confirm decision and the
2249	CONFIRM_TRANSACTION had a "repo	rt-hazards" value of "false".
2250		
	Parameter	Туре

target address

BTP address

	Parameter	Туре		
	address as decider	BTP address		
	transaction-identifier	identifier		
	qualifiers	List of qualifiers		
2251 2252 2253 2254 2255 2256	sent., this will be the reply message.	s to which the TRANSACTION_CONFIRMED is address from the CONFIRM_TRANSACTION		
2257 2258 2259		ion identifier, this determines who the message is		
2260 2261 2262	transaction identifier the the identifier of the Decide	transaction identifier as on the BEGUN message (i.e. or as a whole).		
2263 2264	qualifiers standardised or	other qualifiers.		
2265 2266	Types of FAULT possible (sent to add	ress-as-decider)		
22002267General2268InvalidTerminator – if Terminator address is unknown2269UnknownTransaction – if the transaction-identifier is unknown2270				
2270 2271 2272	CANCEL_TRANSACTION	TRANSACTION		
2273 2274 2275	Sent by a Terminator to a Decider at an sent.	Ferminator to a Decider at any time before CONFIRM_TRANSACTION has been		
	Parameter	Туре		
	target address	BTP address		
	reply address	BTP address		
	transaction identifier	Identifier		
	report-hazard	Boolean		
	qualifiers	List of qualifiers		
2276target address the address to which the CANCEL_TRANSA2278sent. This will be the decider-address from the BEGUN messa2279reply address the address of the Terminator sending the2280CANCEL_TRANSACTION message.2282		er-address from the BEGUN message.		

2283 2284		ntifies the Decider and will be the transaction-identifier	
2284 2285	from the BEGUN message	·.	
2286	report hazard Defines wh	nether the Terminator wishes to be informed of hazard	
2287	•	ecisions within the business transaction. If "report	
2288	hazard" is "true", the recei	ver will wait until responses (CONFIRMED,	
2289		(D) have been received from all of its inferiors,	
2290		vents are reported. If "report hazard" is "false", the	
2291 2292	Decider will reply with TR	RANSACTION_CANCELLED immediately.	
2292	qualifiers standardised or	other qualifiers	
2293	quanters standardised of	oner quamers.	
2295	The business transaction is cancelled –	this is propagated to any remaining Inferiors by	
2296	issuing CANCEL to them. No more In	feriors will be permitted to enrol.	
2297		· · · · ·	
2298	Types of FAULT possible (sent to Sup	erior address)	
2299 2300	General		
2300 2301		f Decider address is unknown	
2302		<i>Ction</i> – if the transaction-identifier is unknown	
2303		CONFIRM_TRANSACTION has been received by	
2304	this Composer.		
2305	-		
2306			
2307	CANCEL_INFERIORS		
2308 2309	Sent by a Terminator to a Decider, but	only if is a Cohesion Composer, at any time before	
2310	CONFIRM_TRANSACTION or CAN		
2311			
	Parameter	Туре	
	target address	BTP address	
	reply address	BTP address	
	transaction identifier	Identifier	
	inferiors-list	List of inferior handles <u>Identifiers</u>	
	qualifiers	List of qualifiers	
2312			
2313	target address the addres	ss to which the CANCEL_TRANSACTION message is	
2314	sent. This will be the decid	sent. This will be the decider-address from the BEGUN message.	
2315			
2316 2317	• •	reply address the address of the Terminator sending the CANCEL_TRANSACTION message.	
2317	CAINCEL_IRAINSAUTIU	n message.	
2318	transaction identifier ide	ntifies the Decider and will be the transaction-identifier	
2320	from the BEGUN message		
	5		

Page 63 of 147

2321		
2322	inferiors-list defines which of	the Inferiors of this Decider are to be cancelled.
2323	using the "inferior-identifiers"	as on the ENROL received by the Decider (in its
2324	role as Superior).	
2325		
2326	qualifiers standardised or oth	er qualifiers.
2327		
2328		
2329	•	s-list are to be cancelled. Any other inferiors are
2330	unaffected by a CANCEL_INFERIORS. F	Further Inferiors may be enrolled.
2331		
0220		all of the automatic encolled Inferiors will
2332 2333		all of the currently enrolled Inferiors will
2333 2334	· · · ·	ermitted to continue with new Inferiors, if
2554	any enrol.	
2335		
2336	Types of FAULT possible (sent to Superior	or address)
2337		
2338	General	
2339		ecider address is unknown
2340		η – if the transaction-identifier is unknown
2341		inferior-handle on the inferiors-list is unknown
2342		NFIRM_TRANSACTION or
2342		CTION has been received by this Composer.
2344		errore has been received by this composer.
2345		
2346		
2347	TRANSACTION_CANCELLED	
2348	-	
2349	A Decider sends TRANSACTION_CANC	CELLED to a Terminator in reply to
2350	REQUEST_CANCEL or in reply to CON	FIRM_TRANSACTION if the Decider decided to
2351	cancel. In both cases, TRANSACTION_C	ANCELLED is used only if all Inferiors cancelled
2352	without reporting hazards or the CANCEL	_TRANSACTION or
2353	CONFIRM_TRANSACTION had a "repo	rt-hazard" value of "false.
2354		
	Parameter	
	target address	BTP address
	address as decider	BTP address
	transaction-identifier	identifier
	qualifiers	List of qualifiers
2355	yuunoi s	

2356	5	lress to which the TRANSACTION_CANCELLED is
2357		ply address from the CANCEL_TRANSACTION or
2358	CONFIRM_TRANSAC	CTION message.
2359		
2360		e address as decider of the Decider as on the BEGUN
2361		action identifier, this determines who the message is
2362	from).	
2363		
2364		the transaction identifier as on the BEGUN message (i.e.
2365	the identifier of the Dec	ider as a whole).
2366		
2367	qualifiers standardised	or other qualifiers.
2368		11 1 1 1
2369	Types of FAULT possible (sent to a	(dress-as-decider)
2370	Comorol	
2371	General	
2372		ator – if Terminator address is unknown
2373	UnknownTrans	saction – if the transaction-identifier is unknown
2374		
2375 2376		
2370	REQUEST_INFERIOR_STATUSES	
2377	REQUEST_INFERIOR_STATUSES	
2379	Sent to a Decider to ask it to report t	the status of its Inferiors with an INFERIOR_STATUSES
2380	-	actor with an address-as-superior or address-as-inferior,
2381		saction tree nodes Inferiors, if there are any. In this latter
2382		uest with a FAULT(StatusRefused). If it is prepared to
2383	reply, but has no Inferiors, it replies	with an INFERIOR_STATUSES with an empty "status-
2384	list" parameter.	
2385		
	Parameter	Туре
	target address	BTP address
	reply address	BTP address
	target-identifier	Identifier
	inferiors-list	List of inferior handlesIdentifiers
	Qualifiers	List of qualifiers
2386		
2387	target address the add	lress to which the REQUEST_STATUS message is sent.
2388	0	r, this will be the address-as-decider from the BEGUN
2389	message. Otherwise it n	nay be an address-as-superior from a CONTEXT or
2390	address-as-inferior from	an ENROL message.
2391		
2392		ress to which the replying INFERIOR_STATUSES is to
2393	be sent	

l

0 00 t		
2394		
2395	0	the transaction (or transaction tree node) within the
2396	· · ·	When the message is used to a Decider, this will be
2397		om the BEGUN message. Otherwise it will be the
2398	-	ONTEXT or an inferior-identifier from an ENROL
2399	message.	
2400	inferieur liet 1 cm 1 1 1	
2401		inferiors enrolled with the target are to be included
2402		SES, using the "inferior-identifiers" as on the ENROL
2403		its role as Superior). If the list is absent, the status of
2404	all enrolled <u>linferiors</u> will b	e reported.
2405		
2406	qualifiers standardised or o	other qualifiers.
2407		
2408	Types of FAULT possible (sent to reply	y-address)
2409		
2410	General	
2411		the receiver is not prepared to report its status to the
2412	0	is FAULT type shall not be issued when a Decider
2413	receives REQUES_STATU.	SES from the Terminator.
2414	UnknownTransaction – if	the transaction-identifier is unknown
2415		
2416		
2417	The form REQUEST_INFERIOR_STA	TUSES/all refers to a REQUEST_STATUS with the
2418	inferiors-list absent. The form REQUES	T_INFERIOR_STATUS/specific refers to a
2419	REQUEST_INFERIOR_STATUS with	the inferiors-list present.
2420		
2421	INFERIOR_STATUSES	
2422		
2423		f all or some of its inferiors in response to a
2424	REQUEST_INFERIOR_STATUSES, F	PREPARE_INFERIORS, CANCEL_INFERIORS,
2425	CANCEL_TRANSACTION with "repo	ort-hazard" value of "true" and
2426		port-hazard"value of "true". It is also used by any
2427	1	ST_INFERIOR_STATUSES to report the status of
2428	inferiors, if there are any.	
2429		
	Parameter	Туре
	target address	BTP address
	responders address	BTP address
	responders-identifier	Identifier
	status-list	Set of Status items - see below
	general-qualifiers	List of qualifiers

2431	target address the address to which the INFERIOR_STATUSES is sent. This	
2432	will be the reply address on the received message	
2433		
2434 2435		he sender is a Decider, the address-as-decider as on the
2435 2436	BEGUN message. Otherwise the address of the sender of this message one of address as inferior, address as superior. With the responders identifier, this	
2430	determines who the messa	•
2438	determines who the messe	
2439	responders-identifier If	the sender is a Decider, the transaction identifier as on
2440	the BEGUN message . Otherwise, the target-identifier used on the	
2441	REQUEST_INFERIOR_S	
2442	-	
2443	status-list contains a num	nber of Status-items, each reporting the status of one of
2444	the inferiors of the Decide	er. The fields of a Status-item are
2445		
	Field	Туре
	Inferior-	Inferioridentifierhandle, identifying which inferior
	handleidentifier	this Status-item contains information for.
	Status	One of the status values below (these are a
		subset of those for STATUS)
	Qualifiers	A list of qualifiers as received from the particular
		inferior or associated with the inferior in earlier
		messages (e.g. an Inferior name qualifier).
2446		
2447	The status value reports the current status of the particular inferior, as known to	
2448	the Decider (Composer or	Coordinator). Values are:
2449	_	
	status value	Meaning
	active	The Inferior is enrolled
	resigned	RESIGNED has been received from the Inferior
	preparing	PREPARE has been sent to the inferior, none of
	1 1 5	PREPARED, RESIGNED, CANCELLED, HAZARD have
		been received
	prepared	PREPARED has been received
	autonomously confirmed	CONFIRMED/auto has been received, no completion
	autonomously commuted	message has been sent
	autonomously assalled	5
	autonomously cancelled	PREPARED had been received, and since then
		CANCELLED has been received but no completion message has been sent
		•
	confirming	CONFIRM has been sent, no outcome reply has been
		received

	status value	Meaning
	confirmed	CONFIRMED/response has been received
	cancelling	CANCEL has been sent, no outcome reply has been received
	cancelled	CANCELLED has been received, and PREPARED was not received previously
	cancel-contradiction	Confirm had been ordered (and may have been sent), but CANCELLED was received
	confirm-contradiction	Cancel had been ordered (and may have been sent) but CONFIRM/auto was received
	hazard	A HAZARD message has been received
	invalid	No such inferior is enrolled (used only in reply to a REQUEST_INFERIOR_STATUSES/specific)
2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469	INFERIOR_STATUSES field containing qualified If the inferiors-list parameter was pro- identified by that parameter shall have the inferiors-list parameter was absent except that an inferior that had been INFERIOR_STATUSES message means Types of FAULT possible (sent to an <i>General</i> <i>InvalidTermina</i>	
2470 2471 2472 2473 2474 2475 2476 2477 2478	group is not just the aggregate of the indicate relatedness. Messages appea indicate messages that may or may n	sages form related groups, for which the meaning of the meanings of the messages. The "&" notation is used to aring in parentheses in the names of groups in this section not be present. The notation A & B / & C in a group name contains A and B or A and C or A, B and C, possibly

2479 CONTEXT & application message

 Meaning: the transmission of the application message is deemed to be part of the business transaction identified by the CONTEXT. The exact effect of this for application work implied by the transmission of the message is determined by the application – in many cases, it will mean the effects of the application message are to be subject to the outcome delivered to an enrolled Inferior, thus requiring the enrolment of a new Inferior if no appropriate Inferior is enrolled or if the CONTEXT is for cohesion.

- **Target address:** the target address is that of the application message. It is not required2489that the application address be a BTP address (in particular, there is no BTP-defined2490"additional information" field the application protocol (and its binding) may or may not2491have a similar construct).
- 2493There may be multiple application messages related to a single CONTEXT message. All2494the application messages so related are deemed to be part of the business transaction2495identified by the CONTEXT. This specification does not imply any further relatedness2496among the application messages themselves (though the application might).
- 2498The actor that sends the group shall retain knowledge of the Superior address in the2499CONTEXT. If the CONTEXT is a CONTEXT/atom, the actor shall also keep track of2500transmitted CONTEXTs for which no CONTEXT_REPLY has been received.
- 2502If the CONTEXT is a CONTEXT/atom, the actor receiving the CONTEXT shall ensure2503that a CONTEXT_REPLY message is sent back to the reply address of the CONTEXT2504with the appropriate completion status.25052505

2506Note – The representation of the relation between CONTEXT and one or2507more application messages depends on the binding to the carrier protocol. It2508is not necessary that the CONTEXT and application messages be closely2509associated "on the wire" (or even sent on the same connection) – some kind2510of referencing mechanism may be used.

2512 CONTEXT_REPLY & ENROL

Meaning: the enrolment of the Inferior identified in the ENROL is to be performed with the Superior identified in the CONTEXT message this CONTEXT_REPLY is replying to. If the "completion-status" of CONTEXT_REPLY is "related", failure of this enrolment shall prevent the confirmation of the business transaction.

- **Target address**: the target address is that of the CONTEXT_REPLY. This will be the2520reply address of the CONTEXT message (in many cases, including request/reply2521application exchanges, this address will usually be implicit).
 - The target address of the ENROL message is omitted.

2524	
2525	The actor receiving the related group will use the retained Superior address from the
2526	CONTEXT sent earlier to forward the ENROL. When doing so, it changes the ENROL to
2527	ask for a response (if it was an ENROL/no-rsp-req) and supplies its own address as the
2528	"reply-address", remembering the original "reply-address" if there was one.
2529	
2530	If ENROLLED is received and the original received ENROL was ENROL/rsp-req, the
2531	ENROLLED is forwarded back to the original "reply-address".
2532	
2533	If this attempt fails (i.e. ENROLLED is not received), and the "completion-status" of the
2534	CONTEXT_REPLY was "related", the actor is required to ensure that the Superior does
2535	not proceed to confirmation. How this is achieved is an implementation option, but must
2536	take account of the possibility that direct communication with the Superior may fail. (One
2537	method is to prevent CONFIRM_TRANSACTION being sent to the Superior (in its role
2538	as Decider); another is to enrol as another Inferior before sending the original CONTEXT
2539	out with an application message). If the Superior is a sub-coordinator or sub-composer,
2540	an enrolment failure must ensure the sub-coordinator does not send PREPARED to its
2541	own Superior.
2542	L L L L L L L L L L L L L L L L L L L
2543	If the actor receiving the related group is also the Superior (i.e. it has the same binding
2544	address), the explicit forwarding of the ENROL is not required, but the resultant effect –
2545	that if enrolment fails the Superior does not confirm or issue PREPARED – shall be the
2546	same.
2547	
2548	A CONTEXT_REPLY & ENROL group may contain multiple ENROL messages, for
2549	several Inferiors. Each ENROL shall be forwarded and an ENROLLED reply received
2550	before the Superior is allowed to confirm if the "completion-status" in the
2551	CONTEXT_REPLY was "related".
2552	
2553	When the group is constructed, if the CONTEXT had "superior-type" value of "atom",
2554	the "completion-status" of the CONTEXT_REPLY shall be "related". If the "superior-
2555	type" was "cohesive", the "completion-status" shall be "completed" or "related" (as
2556	required by the application). If the value is "completed", the actor receiving the group
2557	shall forward the ENROLs, but is not required to (though it may) prevent confirmation.
2558	
2559	CONTEXT_REPLY (& ENROL) & PREPARED / & CANCELLED
2560	- · · /
2561	This combination is characterised by a related CONTEXT_REPLY and either or both of
2562	PREPARED and CANCELLED, with or without ENROL.
2563	
2564	Meaning: If ENROL is present, the meaning and required processing is the same as for
2565	CONTEXT_REPLY & ENROL. The PREPARED or CANCELLED message(s) are
2566	forwarded to the Superior identified in the CONTEXT message this CONTEXT_REPLY
2567	is replying to.
2568	

2569	Note – the combination of CONTEXT_REPLY & ENROL & CANCELLED
2570	may be used to force cancellation of an atom
2571	
2572	Target address: the target address is that of the CONTEXT_REPLY. This will be the
2573	reply address of the CONTEXT message (in many cases, including request/reply
2574	application exchanges, this address will usually be implicit).
2575	
2576	The target address of the PREPARED and CANCELLED message is omitted – they will
2577	be sent to the Superior identified in the earlier CONTEXT message.
2578	
2579	The actor receiving the group forwards the PREPARED or CANCLLED message to the
2580	Superior in as for an ENROL, using the retained Superior address from the CONTEXT
2581	sent earlier, except there is no reply required from the Superior.
2582	
2583	If (as is usual) an ENROL and PREPARED or CANCELLED message are for the same
2584	Inferior, the ENROL shall be sent first, but the actor need not wait for the ENROLLED to
2585	come back before sending the PREPARED or CANCELLED (so an
2586	ENROL+PREPARED bundle from this actor to the Superior could be used).
2580	
2588	The group can contain multiple ENROL, PREPARED and CANCELLED messages.
2589	Each PREPARED and CANCELLED message will be for a different Inferior There is
2590	no constraint on the order of their forwarding, except that ENROL and PREPARED or
2591	CANCELLED for the same Inferior shall be delivered to the Superior in the order
2592	ENROL first, followed by the other message for that Inferior.
2593	
2594	
2595	
2596	CONTEXT_REPLY & ENROL & application message (& PREPARED)
2597	·····=···=··
2598	The presence and details of this section are part of the proposed solution to issue 82,
2599	which was discussed at the BTP committee conference call on 16 Jaunary 2002, but
2600	for which decision was deferred. Accordingly it may be modified or removed when
2601	issue 82 is finalised.
2602	
2602	This combination is characterised by a related CONTEXT_REPLY, ENROL and an
2603	application message. PREPARED may or may not be present in the related group.
2605	approaction message. Their their may of may not be present in the related group.
2605	Meaning: the relation between the BTP messages is as for the preceding groups, The
2600	transmission of the application message (and application effects implied by its
2608	transmission) has been associated with the Inferior identified by the ENROL and will be
2609	subject to the outcome delivered to that Inferior.
2610	
2610	Target address: the target address of the group is the target address of the
2612	CONTEXT_REPLY which shall also be the target address of the application message.
2612	The ENROL and PREPARED messages do not contain their target addresses.

2614 2615 2616 2617 2618 2619 2620 2621	The processing of ENROL and PREPARED messages is the same as for the previous groups. This group can be used when participation in business transaction (normally a cohesion), is initiated by the service (Inferior) side, which fetches or acquires the CONTEXT, with some associated application semantic, performs some work for the transaction and sends an application message with a related ENROL. The CONTEXT_REPLY allows the
2622 2623 2624	addressing of the application (and the CONTEXT_REPLY) to be distinct from that of the Superior.
2625 2626 2627	The actor receiving the group may associate the "inferior-handleidentifier" received on the ENROLLED with the application message in a manner that is visible to the application receiving the message (e.g. for subsequent use in -Terminator:Decider
2628 2629 2630	exchanges). BEGUN & CONTEXT
2631 2632	Meaning: the CONTEXT is that for the new business transaction, containing the
2633 2634 2635	Superior address. Target address: the target address is that of the BEGUN message – this will be the reply
2636 2637 2638	address of the earlier BEGIN message. BEGIN & CONTEXT
2639 2640 2641 2642 2643	Meaning : the new business transaction is to be an Inferior (sub-coordinator or sub- composer) of the Superior identified by the CONTEXT. The Factory (receiver of the BEGIN) will perform the enrolment.
2644 2645 2646	Target address: the target address is that of the BEGIN – this will be the address of the Factory.
2647 2648	Standard qualifiers
2649 2650 2651 2652 2653	The following qualifiers are expected to be of general use to many applications and environments. The URI "urn:oasis:names:tc:BTP:qualifiers" is used in the Qualifier group value for the qualifiers defined here.
2655 2655 2655	Transaction timelimit
2656 2657 2658 2659 2660	The transaction timelimit allows the Superior (or an application element initiating the business transaction) to indicate the expected length of the active phase, and thus give an indication to the Inferior of when it would be appropriate to initiate cancellation if the active phase appears to continue too long. The time limit ends (the clock stops) when the Inferior decides to be prepared and issues PREPARED to the Superior.
	and a prepared and issues i real inclusion of the puperion

0.001		
2661		, the limit has not show of the new install states of
2662		e time limit does not change the permissible actions of
2663		ng to be prepared (for an Inferior), the Inferior is
2664	-	nternal reasons. The timelimit gives an indication to the
2665	entity of when it will be useful to exerc	cise this right.
2666		
2667	The qualifier is propagated on a CONT	EXT message.
2668		
2669	The "Qualifier name" shall be "trans	action-timelimit".
2670		
2671	The "Content" shall contain the follow	ing field:
2672		
	Content field	Туре
	Timelimit	Integer
2673		
2674	Timelimit indicates the maximum (furt	her) duration, expressed as whole seconds from the
2675		CONTEXT, of the active phase of the business
2676	transaction.	
2677		
2678	Inferior timeout	
2679		
2680	This qualifier allows an Inferior to limi	t the duration of its "promise", when sending
2681		bility to confirm or cancel the effects of all associated
2682		ferior is expected to retain the ability to confirm or
2682		expire, the Inferior is released from its promise and
2684	can apply the decision indicated in the	
2685	can apply the decision materice in the	quantient
2686	It should be noted that BTP recognises	the possibility that an Inferior may be forced to apply
2687		CONFIRM or CANCEL is received and before this
2688		ot used). Such a decision is termed a heuristic decision,
2689		sms), is considered to be an exceptional event. As with
2690		onomous decision by a Inferior subsequent to the
2690	•	e contradictory decisions across the business
2692	x	ne occurrence of such a contradiction will be
2692		the business transaction. BTP treats "true" heuristic
2693		ter timeout the same way $-$ in fact, the expiry in this
2695		ate table) change in what can happen, but rather a step
2696	change in the probability that it will.	ate table) change in what can happen, but rather a step
2697	change in the probability that it will.	
2698	The expiry of the timeout does not strid	ctly require that the Inferior immediately invokes the
2699		to do so. An implementation may choose to only
2700		for the underlying resource, for example.
2700		ded to avoid relying on this and ensure decisions for
2701 2702		re these timeouts expire (and allow a margin of error
2702	for network latency etc.).	te these timeouts expire (and anow a margin of enor
2703	tor network fatchey etc.).	
2704		

- The qualifier may be present on a PREPARED message. If the PREPARED message has the "default is cancel" parameter "true", then the "IntendedDecision" field of this qualifier shall have the value "cancel".
- 2709 The "Qualifier name" shall be "inferior-timeout".
- 2711 The "Content" shall contain the following fields:
 - Content fieldTypeTimeoutIntegerIntendedDecision"confirm" or "cancel"
- 2714**Timeout** indicates how long, expressed as whole seconds from the time of transmission of the2715carrying message, the Inferior intends to maintain its ability to either confirm or cancel the2716effects of the associated operations, as ordered by the receiving Superior.
- IntendedDecision indicates which outcome will be applied, if the timeout completes and an
 autonomous decision is made.

2721 Minimum inferior timeout

- This qualifier allows a Superior to constrain the Inferior timeout qualifier received from the
 Inferior. If a Superior knows that the decision for the business transaction will not be
 determined for some period, it can require that Inferiors do not send PREPARED messages
 with Inferior timeouts that would expire before then. An Inferior that is unable or unwilling to
 send a PREPARED message with a longer (or no) timeout should cancel, and reply with
 CANCELLED.
- The qualifier may be present on a CONTEXT, ENROLLED or PREPARE message. If
 present on more than one, and with different values of the MinimumTimeout field, the value
 on ENROLLED shall prevail over that on CONTEXT and the value on PREPARE shall
 prevail over either of the others.
- 2735 The "Qualifier name" shall be "minimum-inferior-timeout".
- 2737 The "Content" shall contain the following field:

Content fieldTypeMinimumTimeoutInteger

- 2740 Minimum Timeout is the minimum value of timeout, expressed as whole seconds, that will be
 2741 acceptable in the Inferior timeout qualifier on an answering PREPARED message.
 2742
- 2743 Inferior name
- 2744

2708

2710

2712

2713

2717

2722

2729

2734

2736

2738

2745 2746 2747 2748	This qualifier allows an Enroller to supply a name for the Inferior that will be visible on INFERIOR_STATUSES and thus allow the Terminator to determine which Inferior (of the Composer or Coordinator) is related to which application work. This is in addition to the "inferior-identifier-handle" field. The name can be human-readable and can also be used in	
2749	fault tracing, debugging and auditing.	
2750		
2751	The name is never used by the BTP actors t	hemselves to identify each other or to direct
2752		es and the identifiers in the message parameters
2753	for those purposes.)	
2754		
2755	This specification makes no requirement that	at the names are unambiguous within any scope
2756		handleidentifier" on ENROLLED and BEGUN,
2757		the scope of the Decider). Other specifications,
2758		particular application may place requirements on
2759	the use and form of the names. (This may include reference to information passed in	
2760	application messages or in other, non-stand	ardised, qualifiers.)
2761		
2762		NROL and in the "qualifiers" field of a Status-item
2763		BEGIN only if there is a related CONTEXT; if
2764	present, the same qualifier value should be	
2765	INFERIOR_STATUSES includes a Status-	
2766	inferior-name qualifier, the same qualifier v	alue should be included in the Status-item.
2767		
2768	The "Qualifier -name" shall be "inferior	-name"
2769	The "Content" shall contain the following f	-1.d.,
2770	The "Content" shall contain the following f	leids:
2771		
	Content field	Туре
	inferior-name	String
2772		

2773 Inferior name the name assigned to the enrolling Inferior.2774

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

2775 State Tables

- Explanation of the state tables
 The state tables deal with the state transitions of the Superior and Inferior roles and which
 message can be sent and received in each state. The state tables directly cover only a single,
 bi-lateral Superior:Inferior relationship. The interactions between, for example, multiple
 Inferiors of a single Superior that will apply the same decision to all or some (of them, are
 dealt with in the definitions of the "decision" events which also specify when changes are
- 2783 2784

2790

2814

2815 2816

There are two state tables, one for Superior, one for Inferior. States are identified by a letterdigit pair, with upper-case letters for the superior, lower-case for the inferior. The same letter is used to group states which have the same, or similar, persistent state, with the digit indicating volatile state changes or minor variations. Corresponding upper and lower-case letters are used to identify (approximately) corresponding Superior and Inferior states.

made to persistent state information (see below).

The Inferior table includes events occurring both at the Inferior as such and at the associated
Enroller, as the Enroller's actions are constrained by and constrain the Inferior role itself.

2794 Status queries

2795 2796 In BTP the messages SUPERIOR STATE and INFERIOR STATE are available to prompt 2797 the peer to report its current state by repeating the previous message (when this is allowed) or 2798 by sending the other * STATE message. The "reply requested" parameter of these messages 2799 distinguishes between their use as a prompt and as a reply. An implementation receiving a *_STATE message with "reply_requested" as "true" is not required to reply immediately – it 2800 may choose to delay any reply until a decision event occurs and then send the appropriate 2801 2802 new message (e.g. on receiving INFERIOR STATE/prepared/y while in state E1, a superior is permitted to delay until it has performed "decide to confirm" or "decide to cancel"). 2803 2804 However, this may cause the other side to repeatedly send interrogatory * STATE messages. 2805

2806 Note that a Superior (or some entity standing in for a now-extinct Superior) uses 2807 SUPERIOR_STATE/unknown to reply to messages received from an Inferior where the 2808 Superior:Inferior relationship is in an unknown (using state "Y1"). The *_STATE messages with a "state" value "inaccessible" can be used as a reply when **any** message is received and 2809 2810 the implementation is temporarily unable to determine whether the relationship is known or what the state is. Other than these cases, the * STATE messages with "reply requested" equal 2811 2812 to "false" are only sent when the other message with "reply requested" equal to "true" has 2813 been received and no other message has been sent.

Decision events

The persistent state changes (equivalent to logging in a regular transaction system) and some other events are modelled as "decision events" (e.g. "decide to confirm", "decide to be prepared"). The exact nature of the real events and changes in an implementation that are modelled by these events depends on the position of the Superior or Inferior within the

- business transaction and on features of the implementation (e.g. making of a persistent record of the decision means that the information will survive at least some failures that otherwise
 lose state information, but the level of survival depends on the purpose of the implementation). Table 2Table 2 and Table 3 Table 3 define the decision events.
- In some cases, an implementation may not need to make an active change to have a persistent record of a decision, provided that the implementation will restore itself to the appropriate state on recovery. For example, an (inferior) implementation that "decided to be prepared", and recorded a timeout (to cancel) in the persistent information for that decision (signalled via the appropriate qualifier on PREPARED), could treat the presence of an expired record as a record of "decide to cancel autonomously", provided it always updated such a record as part of the "apply ordered confirmation" decision event.
- The Superior event "decide to prepare" is considered semi-persistent. Since the sending of 2834 2835 PREPARE indicates that the application exchange (to associate operations with the Inferior) is complete, it is not meaningful for the Superior:Inferior relationship to revert to an earlier 2836 2837 state corresponding to an incomplete application exchange. However, implementations are not required to make the sending of PREPARE persistent in terms of recovery – a Superior 2838 that experiences failure after sending PREPARE may, on recovery, have no information 2839 2840 about the transaction, in which case it is considered to be in the completed state (Z), which will imply the cancellation of the Inferior and its associated operations. 2841
- Where a Superior is itself an Inferior (to another Superior entity), in a hierarchic tree, its
 "decide to confirm" and "decide to cancel" decisions will in fact be the receipt of a
 CONFIRM or CANCEL instruction from its own Superior, without necessary change of local
 persistent information (which would combine both superior and inferior information, pointing
 both up and down the tree).
- 2848 2849

2857

2842

2825

2833

2850 **Disruptions – failure events**

Failure events are modelled as "disruption". A failure and the subsequent recovery will (or may) cause a change of state. The disruption events in the state tables model different extents of loss of state information. An implementation is not required to exhibit all the possible disruption events, but it is not allowed to exhibit state transitions that do not correspond to a possible disruption.

In addition to the disruption events in the tables, there is an implicit "disruption 0" event,
which involves possible interruption of service and loss of messages in transit, but no change
of state (either because no state information was lost, or because recovery from persistent
information restores the implementation to the same state). The "disruption 0" event would
typically be an appropriate abstraction for a communication failure.

2863

2864 Invalid cells and assumptions of the communication mechanism 2865

2866The empty cells in state table represent events that cannot happen. For events corresponding2867to sending a message or any of the decision events, this prohibition is absolute – e.g. a

I

2868 2869 2870	conformant implementation in the Superior active state "B1" will not send CONFIRM. For events corresponding to receiving a message, the interpretation depends on the properties of the underlying communications mechanism.
2871 2872 2873 2874 2875 2876 2877	 For all communication mechanisms, it is assumed that a) the two directions of the Superior:Inferior communication are not synchronised – that is messages travelling in opposite directions can cross each other to any degree; any number of messages may be in transit in either direction; and b) messages may be lost arbitrarily
2877 2878 2879 2880 2881 2882	If the communication mechanisms guarantee ordered delivery (i.e. that messages, if delivered at all, are delivered to the receiver in the order they were sent), then receipt of a message in a state where the corresponding cell is empty indicates that the far-side has sent a message out of order – a FAULT message with the Fault Type "WrongState" can be returned.
2883 2884 2885 2886 2887	If the communication mechanisms cannot guarantee ordered delivery, then messages received where the corresponding cell is empty should be ignored. Assuming the far-side is conformant, these messages can assumed to be "stale" and have been overtaken by messages sent later but already delivered. (If the far-side is non-conformant, there is a problem anyway).
2888 2889	Meaning of state table events
2890 2891 2892 2893 2894 2895	The tables in this section define the events (rows) in the state tables. <u>Table 1</u> Table 1 defines the events corresponding to sending or receiving BTP messages and the disruption events. <u>Table 2</u> describes the decision events for an Inferior, <u>Table 3</u> Table 3 those for a Superior.
2896 2897 2898	The decision events for a Superior, defined in <u>Table 3 Table 3</u> cannot be specified without reference to other Inferiors to which it is Superior and to its relation with the application or other entity that (acting ultimately on behalf of the application) drives it.
2899 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909 2910 2911	The term "remaining Inferiors" refers to any actors to which this endpoint is Superior and which are to be treated as an atomic decision unit with (and thus including) the Inferior on this relationship. If the CONTEXT for this Superior:Inferior relationship had a "superior type" of "atom", this will be all Inferiors established with same Superior address and Superior identifier except those from which RESIGN has been received. If the CONTEXT had "superior type" of "cohesion", the "remaining Inferiors" excludes any that it has been determined will be cancelled, as well as any that have resigned – in other words it includes only those for which a confirm decision is still possible or has been made. The determination of exactly which Inferiors are "remaining Inferiors" in a cohesion is determined, in some way, by the application. The term "Other remaining Inferiors" excludes this Inferior on this relationship. A Superior with a single Inferior will have no "other remaining Inferiors".
2912 2913 2914	In order to ensure that the confirmation decision is delivered to all remaining Inferiors, despite failures, the Superior must persistently record which these Inferiors are (i.e. their addresses and identifiers). It must also either record that the decision is confirm, or ensure

I

l

I

2915 that the confirm decision (if there is one) is persistently recorded somewhere else, and that it will be told about it. This latter would apply if the Superior were also BTP Inferior to another 2916 entity which persisted a confirm decision (or recursively deferred it still higher). However, 2917 since there is no requirement that the Superior be also a BTP Inferior to any other entity, the 2918 behaviour of asking another entity to make (and persist) the confirm decision is termed 2919 2920 "offering confirmation" - the Superior offers the possible confirmation of itself, and its 2921 remaining Inferiors to some other entity. If that entity (or something higher up) then does make and persist a confirm decision, the Superior is "instructed to confirm" (which is 2922 equivalent BTP CONFIRM). 2923 2924

The application, or an entity acting indirectly on behalf of the application, may request a Superior to prepare an Inferior (or all Inferiors). This typically implies that there will be no more operations associated with the Inferior. Following a request to prepare all remaining Inferiors, the Superior may offer confirmation to the entity that requested the prepare. (If the Superior is also a BTP Inferior, its superior can be considered an entity acting on behalf of the application.)

2932The application, or an entity acting indirectly on behalf of the application, may also request2933confirmation. This means the Superior is to attempt to make and persist a confirm decision2934itself, rather than offer confirmation.

- 2935 2936
- 2750
- 2937

Table 1 : send, receive and disruption events

Event name	Meaning
send/receive ENROL/rsp-req	send/receive ENROL with reply-requested = true
send/receive ENROL/no-rsp-req	send/receive ENROL with reply-requested = false
send/receive RESIGN/rsp-req	send/receive RESIGN with reply-requested = true
send/receive RESIGN/no-rsp-req	send/receive RESIGN with reply-requested = false
send/receive PREPARED	send/receive PREPARED, with default-cancel = false
send/receive PREPARED/cancel	send/receive PREPARED, with default-cancel = true
send/receive CONFIRMED/auto	send/receive CONFIRMED, with confirm-received = true
send/receive CONFIRMED/response	send/receive CONFIRMED, with confirm-received = false
send/receive HAZARD	send/receive HAZARD
send/receive INF_STATE/***/y	send/receive INFERIOR_STATE with status *** and reply-requested = true
send/receive INF_STATE/***	send/receive INFERIOR_STATE with status *** and reply-requested = false

Event name	Meaning
send/receive SUP_STATE/***/y	send/receive SUPERIOR_STATE with status *** and reply-requested = true ("prepared-rcvd" represents "prepared-received")
send/receive SUP_STATE/***	send/receive SUPERIOR_STATE with status *** and reply-requested = false ("prepared-rcvd" represents "prepared-received")
disruption ***	Loss of state– new state is state applying after any local recovery processes complete

2939

Table 2 : Decision events for Inferior

Event name	Meaning
decide to resign	 Any associated operations have had no effect (data state is unchanged)).
decide to be prepared	Effects of all associated operations can be confirmed or cancelled;
	 information to retain confirm/cancel ability has been made persistent
decide to be prepared/cancel	As "decide to be prepared";
	 the persistent information specifies that the default action will be to cancel
decide to confirm autonomously	 Decision to confirm autonomously has been made persistent;
	 the effects of associated operations will be confirmed regardless of failures
decide to cancel autonomously	 Decision to cancel autonomously has been made persistent
	 the effects of associated operations will be cancelled regardless of failures
apply ordered confirmation	 Effects of all associated operations have been confirmed;
	Persistent information is effectively removed
remove persistent information	Persistent information is effectively removed;

Event name	Meaning
detect problem	 For at least some of the associated operations, EITHER they cannot be consistently cancelled or consistently confirmed; OR
	o it cannot be determined whether they will be cancelled or confirmed
	AND, information about this is not persistent
detect and record problem	 As for the first condition of "detect problem" information recording this has been persisted (to the degree considered appropriate), or the detection itself is
	persistent. (i.e. will be re-detected on recovery)

2941

Table 3: Decision events for a Superior

Event name	Meaning
decide to confirm one-phase	 All associated application messages to be sent to the service have been sent;
	There are no other remaining Inferiors
	 If an atom, all enrolments that would create other Inferiors have completed (no outstanding CONTEXT_REPLYs)
	The Superior has been requested to confirm
decide to prepare	 All associated application messages to be sent to the service have been sent;
	• The Superior has been requested to prepare this Inferior
decide to confirm	• Either
	 PREPARED or PREPARED/cancel has been received from all other remaining Inferiors; AND
	o Superior has been requested to confirm; AND
	 persistent information records the confirm decision and identifies all remaining Inferiors;
	• Or
	o persistent information records an offer of confirmation and has been instructed to confirm
decide to cancel	Superior has not offered confirmation; OR
	 Superior has offered confirmation and has been instructed to cancel; OR

Event name	Meaning
	 Superior has offered confirmation but has made an autonomous cancellation decision
remove confirm information	• Persistent information has been effectively removed;
record contradiction	 Information recording the contradiction has been persisted (to the degree considered appropriate)

2944

2951

2957

2963

2943 Persistent information

Persisted information (especially prepared information at an Inferior, confirm information at a
Superior) may include qualifications of the state carried in Qualifiers of the corresponding
message (e.g. inferior timeouts in prepared information). It may also include applicationspecific information (especially in Inferiors) to allow the future confirmation or cancellation
of the associated operations. In some cases it will also include information allowing an
application message sent with a BTP message (e.g. PREPARED) to be repeated.

2952 The "effective" removal of persistent information allows for the possibility that the 2953 information is retained (perhaps for audit and tracing purposes) but some change to the 2954 persistent information (as a whole) means that if there is a failure after such change, on 2955 recovery, the persistent information does not cause the endpoint to return the state it would 2956 have recovered to before the change.

In all cases, the degree to which information described as "persistent" will survive failure is a configuration and implementation option. An implementation **should** describe the level of failure that it is capable of surviving. For applications manipulating information that is itself volatile (e.g. network configurations), there is no requirement to make the BTP state information more persistent that than the application information.

The degree of persistence of the recording of a hazard (problem) at an Inferior and recording of a detected contradiction at a Superior may be different from that applying to the persistent prepared and confirm information. Implementations and configuration may choose to pass hazard and contradiction information via management mechanisms rather than through BTP. Such passing of information to a management mechanism could be treated as "record problem" or "record contradiction".

Table 4 : Superior states

State	summary
11	CONTEXT created
A1	ENROLing
B1	ENROLLED (active)
C1	resigning
D1	PREPARE sent
E1	PREPARED received
E2	PREPARED/cancel received
F1	CONFIRM sent
F2	completed after confirm
G1	cancel decided
G2	CANCEL sent
G3	cancelling, RESIGN received
G4	both cancelled
H1	inferior autonomously confirmed
J1	Inferior autonomously cancelled
K1	confirmed, contradiction detected
L1	cancelled, contradiction detected
P1	hazard reported
P2	hazard reported in null state
P3	hazard reported after confirm decision
P4	hazard reported after cancel decision
Q1	contradiction detected in null state
R1	Contradiction or hazard recorded
R2	completed after contradiction or hazard recorded
S1	one-phase confirm decided
Y1	completed queried
Z	completed and unknown

 Table 5 : Inferior states

<u>.</u>	
State	summary
i1	aware of CONTEXT
a1	enrolling
b1	enrolled
c1	resigning
d1	preparing
e1	prepared
e2	prepared,default to cancel
f1	confirming
f2	confirming after default cancel
g1	CANCEL received in prepared state
g2	CANCEL received in prepared/cancel state
h1	Autonomously confirmed
h2	autonomously confirmed, superior confirmed
j1	autonomously cancelled
j2	autonomously cancelled, superior cancelled
k1	autonomously cancelled, contradicted
k2	autonomously cancelled, CONTRADICTION received
1	autonomously confirmed, contradicted
12	autonomously confirmed, CONTRADICTION received
m1	confirmation applied
n1	cancelling
p1	hazard detected, not recorded
p2	hazard detected in prepared state, not recorded
q1	hazard recorded
s1	CONFIRM_ONE_PHASE received after prepared state
s2	CONFIRM_ONE_PHASE received
s3	CONFIRM_ONE_PHASE received, confirming
s4	CONFIRM_ONE_PHASE received, cancelling
s5	CONFIRM_ONE_PHASE received, hazard detected
s6	CONFIRM_ONE_PHASE received, hazard recorded
x1	completed, presuming abort
x2	completed, presuming abort after prepared/cancel

State	summary
y1	completed, queried
у2	completed, default cancel, a message received
Z	completed
z1	completed with default cancel

 Table 6: Superior state table – normal forward progression

		1		1				1	
	11	A1	B1	C1	D1	E1	E2	F1	F2
receive ENROL/rsp-req	A1								
receive ENROL/no-rsp-req	B1								
receive RESIGN/rsp-req	Y1		C1	C1	C1				
receive RESIGN/no-rsp-req	Ζ		Ζ	Ζ	Ζ				
receive PREPARED	Y1		E1		E1	E1		F1	
receive PREPARED/cancel	Y1		E2		E2		E2	F1	
receive CONFIRMED/auto	Q1		H1		H1	H1		F1	
receive CONFIRMED/response								F2	F2
receive CANCELLED	Y1		Ζ		Ζ	J1	J1	K1	
receive HAZARD	P1	P1	P1		P1	P1	P1	P3	
receive INF_STATE/active/y	Y1	A1	B1		D1				
receive INF_STATE/active			B1		D1				
receive INF_STATE/unknown			Ζ	Ζ	Ζ				
send ENROLLED		B1							
send RESIGNED				Ζ					
send PREPARE					D1	E1	E2		
send CONFIRM_ONE_PHASE									
send CONFIRM								F1	
send CANCEL									
send CONTRADICTION									
send SUP_STATE/active/y			B1						
send SUP_STATE/active			B1						
send SUP_STATE/prepared-rcvd/y						E1	E2		
send SUP_STATE/prepared-rcvd						E1	E2		
send SUP_STATE/unknown									
decide to confirm one-phase			S1			S1	S1		
decide to prepare			D1						
decide to confirm						F1	F1		
decide to cancel			G1		G1	G1	Ζ		
remove persistent information									Ζ
record contradiction									
disruption I	Ζ	Ζ	Ζ	Ζ	Ζ	Ζ	Ζ		F1
disruption II						D1	D1		
disruption III						B1	B1		
disruption IV									

	G1	G2	G3	G4	H1	J1	K1	L1
receive ENROL/rsp-req								
receive ENROL/no-rsp-req								
receive RESIGN/rsp-req	G3	Ζ	G3					
receive RESIGN/no-rsp-req	Ζ	Ζ	Ζ					
recei ve PREPARED	G1	G2						
recei ve PREPARED/cancel	G1	G2						
receive CONFIRMED/auto	L1	L1			H1			L1
receive CONFIRMED/response								
receive CANCELLED	G4	Ζ		G4		J1	K1	
recei ve HAZARD	P4	P4						
receive INF_STATE/active/y	G1	G2						
receive INF_STATE/active	G1	G2						
receive INF_STATE/unknown	Ζ	Ζ	Ζ	Ζ				
send ENROLLED								
send RESIGNED								
send PREPARE								
send CONFIRM_ONE_PHASE								
send CONFIRM								
send CANCEL	G2	G2	Ζ	Ζ				
send CONTRADICTION								
send SUP_STATE/active/y								
send SUP_STATE/active								
send SUP_STATE/prepared-rcvd/y								
send SUP_STATE/prepared-rcvd								
send SUP_STATE/unknown								
decide to confirm one-phase								
decide to prepare								
decide to confirm					F1	K1		
decide to cancel					L1	G4		
remove persistent information								
record contradiction							R1	R1
disruption I	Ζ	Ζ	Ζ	Ζ	Ζ	Ζ	F1	Ζ
disruption II			G2	G2	E1	E1		G2
disruption III					D1	D1		
disruption IV					B1	B1		

 Table 7: Superior state table – cancellation and contradiction

Table 8: Superior state table – hazard and request confirm

	P1	P2	P3	P4	Q1	R1	R2	S1
receive ENROL/rsp-req								
receive ENROL/no-rsp-req								
receive RESIGN/rsp-req								C1
receive RESIGN/no-rsp-req								Ζ
recei ve PREPARED								S1
receive PREPARED/cancel								S1
receive CONFIRMED/auto					Q1	R1	R1	S1
receive CONFIRMED/response					Ζ	R2		Ζ
receive CANCELLED						R1	R1	Ζ
receive HAZARD	P1	P2	P3	Ρ4		R1	R1	Ζ
receive INF_STATE/active/y								S1
receive INF_STATE/active								S1
receive INF_STATE/unknown	P1	P2		Ρ4		R2	R2	Ζ
send ENROLLED								
send RESIGNED								
send PREPARE								
send CONFIRM_ONE_PHASE								S1
send CONFIRM								
send CANCEL								
send CONTRADICTION						R2		
send SUP_STATE/active/y								
send SUP_STATE/active								
send SUP_STATE/prepared-rcvd/y								
send SUP_STATE/prepared-rcvd								
send SUP_STATE/unknown								
decide to confirm one-phase								
decide to prepare								
decide to confirm								
decide to cancel								
remove persistent information							Ζ	
record contradiction	R1	R1	R1	R1	R1			
disruption I	Ζ	Ζ	Ζ	Ζ	Z		R1	Z
disruption II	D1		F1	G2				
disruption III	B1							
disruption IV								

Table 9: Superior state table – query after completion and completed states

	-	
	Y1	Ζ
receive ENROL/rsp-req		Y1
receive ENROL/no-rsp-req		Y1
receive RESIGN/rsp-req	Y1	Y1
receive RESIGN/no-rsp-req	Ζ	Ζ
receive PREPARED	Y1	Y1
receive PREPARED/cancel	Y1	Y1
receive CONFIRMED/auto	Q1	Q1
receive CONFIRMED/response	Ζ	Ζ
receive CANCELLED	Y1	Y1
receive HAZARD	P2	P2
receive INF_STATE/active/y	Y1	Y1
receive INF_STATE/active	Y1	Ζ
receive INF_STATE/unknown	Ζ	Ζ
send ENROLLED		
send RESIGNED		
send PREPARE		
send CONFIRM_ONE_PHASE		
send CONFIRM		
send CANCEL		
send CONTRADICTION		
send SUP_STATE/active/y		
send SUP_STATE/active		
send SUP_STATE/prepared-rcvd/y		
send SUP_STATE/prepared-rcvd		
send SUP_STATE/unknown	Ζ	
decide to confirm one-phase		
decide to prepare		
decide to confirm		
decide to cancel		
remove persistent information		
record contradiction		
disruption I	Ζ	
disruption II		
disruption III		

2981

Table 10: Inferior state table – normal forward progression

		1						1	
1	i1	a1	b1	c1	d1	e1	e2	f1	f2
send ENROL/rsp-req	a1								
send ENROL/no-rsp-req	b1								
send RESIGN/rsp-req				c1					
send RESIGN/no-rsp-req				Z					
send PREPARED						e1			
send PREPARED/cancel							e2		
send CONFIRMED/auto									
send CONFIRMED/response									
send CANCELLED			Z		Z				
send HAZARD									
send INF_STATE/active/y		a1	b1		d1				
send INF_STATE/active			b1		d1				
send INF_STATE/unknown									
receive ENROLLED		b1							
receive RESIGNED				Z					
recei ve PREPARE		d1	d1	c1	d1	e1	e2		
receive CONFIRM_ONE_PHASE		s2	s2	c1		s1	s1		
receive CONFIRM						f1	f2	f1	f2
receive CANCEL		n1	n1	Z	n1	g1	g2		
receive CONTRADICTION									
receive SUP_STATE/active/y		b1	b1	c1		e1	e2		
receive SUP_STATE/active		b1	b1	c1		e1	e2		
recei ve SUP_STATE/prepared-rcvd/y						e1	e2		
receive SUP_STATE/prepared-rcvd						e1	e2		
receive SUP_STATE/unknown		Z	Z	Z	Z	x1	x2		
decide to resign			c1		c1				
decide to be prepared			e1		e1				
decide to be prepared/cancel			e2		e2				
decide to confirm autonomously						h1			
decide to cancel autonomously						j 1	z1		
apply ordered confirmation								m1	m1
remove persistent information									
detect problem		р1	р1		р1	p2	р2	p2	р2
detect and record problem									
disruption I		Z	Z	Z	Z			e1	e2
disruption II					b1				
disruption III									

Table 11: Inferior state table – cancellation and contradiction

	g1	g2	h1	h2	j 1	j 2	k1	k2	11	12
send ENROL/rsp-req										
send ENROL/no-rsp-req										
send RESIGN/rsp-req										
send RESIGN/no-rsp-req										
send PREPARED										
send PREPARED/cancel										
send CONFIRMED/auto			h1						11	
send CONFIRMED/response										
send CANCELLED					j 1		k1			
send HAZARD										
send INF_STATE/active/y										
send INF_STATE/active										
send INF_STATE/unknown										
receive ENROLLED										
receive RESIGNED										
recei ve PREPARE			h1		j 1					
receive CONFIRM_ONE_PHASE			s3		s4					
receive CONFIRM			h2	h2	k1		k1			
receive CANCEL	g1	g2	11		j 2	j 2			11	
receive CONTRADICTION			12		k2		k2	k2	12	12
receive SUP_STATE/active/y			h1		j 1					
receive SUP_STATE/active			h1		j 1					
receive SUP_STATE/prepared-rcvd/y			h1		j 1					
receive SUP_STATE/prepared-rcvd			h1		j 1					
recei ve SUP_STATE/unknown	x1	x2	11		j 2	j 2	k2	k2	11	
decide to resign										
decide to be prepared										
decide to be prepared/cancel										
decide to confirm autonomously										
decide to cancel autonomously										
apply ordered confirmation										
remove persistent information	n1	n1		m1		Ζ		Z		Z
detect problem	p2	р2								
detect and record problem										
disruption I	e1	e2		h1		j 1	j 1	k1	h1	11
disruption II								j 1		h1
disruption III										

Table 12: Inferior state table – confirm, cancel ordered and hazard recording

	m1	n1	p1	p2	q1
send ENROL/rsp-req					
send ENROL/no-rsp-req					
send RESIGN/rsp-req					
send RESIGN/no-rsp-req					
send PREPARED					
send PREPARED/cancel					
send CONFIRMED/auto					
send CONFIRMED/response	Z				
send CANCELLED		Z			
send HAZARD			р1	р2	q1
send INF_STATE/active/y					
send INF_STATE/active					
send INF_STATE/unknown					
receive ENROLLED			p1		q1
receive RESIGNED					
recei ve PREPARE			р1	р2	q1
receive CONFIRM_ONE_PHASE			s5	s5	s6
receive CONFIRM	m1			р2	q1
receive CANCEL		n1	р1	р2	q1
receive CONTRADICTION			Z	Z	Z
receive SUP_STATE/active/y			p1	p2	q1
receive SUP_STATE/active			р1	р2	q1
receive SUP_STATE/prepared-rcvd/y				p2	q1
receive SUP_STATE/prepared-rcvd				p2	q1
receive SUP_STATE/unknown		Z	p1	р2	q1
decide to resign					
decide to be prepared					
decide to be prepared/cancel					
decide to confirm autonomously					
decide to cancel autonomously					
apply ordered confirmation					
remove persistent information					
detect problem			. 1	. 1	
detect and record problem			q1	q1	
disruption I	Z	Z	Z		
disruption II		d1			
disruption III		b1			

Table 13: Inferior state table –	request confirm states
----------------------------------	------------------------

	s1	s2	s3	s4	s5	s6
send ENROL/rsp-req	31	52	30	57	55	50
send ENROL/no-rsp-req						
send RESIGN/rsp-req						
send RESIGN/no-rsp-req						
send PREPARED						
send PREPARED/cancel						
send CONFIRMED/auto						
send CONFIRMED/response			Z			
send CANCELLED				Z		
send HAZARD					z	Z
send INF_STATE/active/y						
send INF_STATE/active						
send INF_STATE/unknown						
receive ENROLLED						
receive RESIGNED						
recei ve PREPARE						
receive CONFIRM_ONE_PHASE	s1	s2	s3	s4	s5	s6
receive CONFIRM						
receive CANCEL						
receive CONTRADICTION			s3		Z	s6
receive SUP_STATE/active/y						
receive SUP_STATE/active						
receive SUP_STATE/prepared-rcvd/y						
receive SUP_STATE/prepared-rcvd						
receive SUP_STATE/unknown	x1	Z	Ζ	Ζ	Ζ	Ζ
decide to resign						
decide to be prepared						
decide to be prepared/cancel						
decide to confirm autonomously		s3				
decide to cancel autonomously		s4				
apply ordered confirmation						
remove persistent information	s2					
detect problem						
detect and record problem		s6				
disruption I	e1	Z		Z	Z	
disruption II						
disruption III						

Table 14: Inferior state table – completed states (including presume-abort and queried)

	x1	x2	v1	y2	z	z1
send ENROL/rsp-req		~~	יע	<u>y</u> 2	~	21
send ENROL/no-rsp-req						
send RESIGN/rsp-req						
send RESIGN/no-rsp-req						
send PREPARED						
send PREPARED/cancel						
send CONFIRMED/auto						
send CONFIRMED/response						
send CANCELLED				z1		
send HAZARD						
send INF_STATE/active/y						
send INF_STATE/active						
send INF_STATE/unknown			z			
receive ENROLLED					Z	
receive RESIGNED			y1		Z	
recei ve PREPARE			y1	y2	y1	z1
receive CONFIRM_ONE_PHASE			y1	y2	y1	y1
receive CONFIRM				y2	m1	y2
receive CANCEL			y1	Z	y1	y1
receive CONTRADICTION			Z	Z	Z	Z
receive SUP_STATE/active/y			y1	y2	y1	y2
receive SUP_STATE/active			y1	y2	Z	z1
receive SUP_STATE/prepared-rcvd/y				y2		y2
receive SUP_STATE/prepared-rcvd				y2		y2
receive SUP_STATE/unknown	x1	x2	y1	y2	Z	Ζ
decide to resign						
decide to be prepared						
decide to be prepared/cancel						
decide to confirm autonomously						
decide to cancel autonomously						
apply ordered confirmation						
remove persistent information	Z	Ζ				
detect problem						
detect and record problem						
disruption I	e1	e2				
disruption II						
disruption III						

2989

2990

2991 Failure Recovery

2992Types of failure

BTP is designed to ensure the delivery of a consistent decision for a business transaction to
the parties involved, even in the event of failure. Failures can be classified as:

- **Communication failure**: messages between BTP actors are lost and not delivered. BTP assumes the carrier protocol ensures that messages are either delivered correctly (without corruption) or are lost, but does not assume that all losses are reported or that messages sent separately are delivered in the order of sending.
- 3002 3003 3004

3006

3019

3029

2993

2997

2998

2999 3000

3001

3004 3005 **Node failure (system failure, site failure)**: a machine hosting one or more BTP actors stops processing and all its volatile data is lost. BTP assumes a site fails by stopping – it either operates correctly or not at all, it never operates incorrectly.

Communication failure may become known to a BTP implementation by an indication from 3007 3008 the lower layers or may be inferred (or suspected) by the expiry of a timeout. Recovery from a communication failure requires only that the two actors can again send messages to each 3009 3010 other and continue or complete the progress of the business transaction. In the state tables for 3011 the Superior:Inferior relationship, each side is either waiting to make a decision or can send a 3012 message. For some states, the message to be sent is a repetition of a regular message; for 3013 other states, the INFERIOR STATE or SUPERIOR STATE message can be sent, requesting 3014 a response. Thus, following a communication failure, either side can prompt the other to re-3015 establish the relationship. Receiving one of the * STATE messages asking for a response 3016 does not require an immediate response – especially if an implementation is waiting to 3017 determine a decision (perhaps because it is itself waiting for a decision from elsewhere), an 3018 implementation may choose not to reply until it wishes too.

3020 A node failure is distinguished from communication failure because there is loss of volatile state. To ensure consistent application of the decision of a business transaction, BTP requires 3021 3022 that some state information will be persisted despite node failure. Exactly what real events 3023 correspond to node failure but leave the persistent information undamaged is a matter for implementation choice, depending on application requirements; however, for most 3024 3025 application uses, power failure should be survivable (an exception would be if the data 3026 manipulated by the associated operations was volatile). There will always be some level of 3027 event sufficiently catastrophic to lose persistent information and the ability to recoverdestruction of the computer or bankruptcy of the organisation, for example. 3028

Recovery from node failure involves recreating the endpoint in a node that has access to the persistent information for incomplete transactions. This may be a recreation of the original node (including the ability to perform application work) using the same addresses; or there may be a distinct recovery entity, which can access the persistent data, but has a different address; other implementation approaches are possible. Restoration of the endpoint from persistent information will often result in a partial loss of state, relative to the volatile state reached before the failure. This is modelled in the state tables by the "disruption" events. 3037After recovery from node failure, the implementation behaves much as if a communication3038failure had occurred.

- 3040 Persistent information
- 3041

3072

3039

3042BTP requires that some decision events are persisted – that information recording an3043Inferior's decision to be prepared, a Superior's decision to confirm and an Inferior's3044autonomous decision survive failure. Making the first two decisions persistent ensures that a3045consistent decision can be reached for the business transaction and that it is delivered to all3046involved nodes. Requiring an Inferior's autonomous decision to be persistent allows BTP to3047ensure that, if this decision is contradictory (i.e. opposite to the decision at the Superior), the3048contradiction will be reported to the Superior, despite failures.3049

3050 BTP also permits, but does not require, recovery of the Superior: Inferior relationship in the active state (unlike many transaction protocols, where a communication or endpoint failure in 3051 active state would invariably cause rollback of the transaction). Recovery in the active state 3052 3053 may require that the application exchange is resynchronised as well – BTP does not directly support this, but does allow continuation of the business transaction as such. In the state 3054 3055 tables, from some states, there are several levels of disruption, distinguished by which state the implementation transits to – this represents the survival of different extents of state 3056 information over failure and recovery. The different levels of disruption describe legitimate 3057 3058 states for the endpoint to be in after it has recovered - they do not require that all implementations are able to exhibit the appropriate partial loss of state information. 3059 The absence of a destination state for the disruption events means that such a transition is not 3060 legitimate – thus, for example, an Inferior that has decided to be prepared will always recover 3061 to the same state, by virtue of the information persisted in the "decide to be prepared" event. 3062

3063 Apart from the (optional) recovery in active state, BTP follows the well-known presume-3064 abort model - it is only required that information be persisted when decisions are made (and 3065 3066 not, e.g. on enrolment). This means that on recovery, one side may have persistent information but the other does not. This occurs when an Inferior has decided to be prepared 3067 but the Superior never confirmed (so the decision is "presumed" to be cancel), or because the 3068 3069 Superior did confirm, and the Inferior applied the confirm, removed its persistent information but the acknowledgement (CONFIRMED) was never received by the Superior (or, at least, it 3070 still had the persistent information when the failure occurred). 3071

Information to be persisted for an Inferior's "decision to be prepared" must be sufficient to
 re-establish communication with the Superior, to apply a confirm decision and to apply a
 cancel decision. It will thus need to include

- 3076 Inferior identity (this may be an index used to locate the information)
- 3077 Superior address (as on CONTEXT)
- 3078 Superior identifier (as on CONTEXT)
- 3079default-is-cancel value (as on PREPARED)3080

3081The information needed to apply confirm/cancel decisions will depend on the application and3082the associated operations. It may also normally be necessary to persist any qualifiers that

3083	were sent with the PREPARED message or application messages sent with the PREPARED,
3084	since the PREPARED message will be repeated if a failure occurs.
3085	
3086	A Superior must record corresponding information to allow it to re-establish communication
3087	with the Inferior:
3088	Inferior address (as on ENROL)
3089	Inferior identifier (as on ENROL)
3090	
3090	A Superior that is the Decider for the husiness transaction need only persist this information
	A Superior that is the Decider for the business transaction need only persist this information
3092	if it makes a decision to confirm (and this Inferior is in the confirm set, for a Cohesion). A
3093	Superior that is also an Inferior to some other entity (i.e. it is an intermediate in a tree, as
3094	atom in a cohesion, sub-coordinator or sub-composer) must persist this information as
3095	Superior (to this Inferior) as part of the persistent information of its decision to be prepared
3096	(as an Inferior). For such an entity, the "decision to confirm" as Superior is made when (and
3097	if) CONFIRM is received from its Superior or it makes an autonomous decision to confirm. If
3098	CONFIRM is received, the persistent information may be changed to show the confirm
3099	decision, but alternatively, the receipt of the CONFIRM can be treated as the decision itself.
3100	If the persistent information is left unchanged and there is a node failure, on recovery the
3101	entity (as an Inferior) will be in a prepared state, and will rediscover the confirm decision
3102	(using the recovery exchanges to its Superior) before propagating it to its Inferior(s).
3103	
3104	After failure, an implementation may not be able to restore an endpoint to the appropriate
3105	state immediately – in particular, the necessary persistent information may be inaccessible,
3106	although the implementation can respond to received BTP messages. In such a case, a
3107	Superior may reply to any BTP message except INFERIOR_STATE/* (i.e. with a "reply-
3107	requested" value "false") with SUPERIOR_STATE/inaccessible and an Inferior to any BTP
3109	message except SUPERIOR_STATE/* with "INFERIOR_STATE/inaccessible. Receipt of
3109	
	the *_STATE/inaccessible messages has no effect on the endpoint state.
3111	
3112	Redirection
3113	
3114	As described above, BTP uses the presume-abort model for recovery. A corollary of this is
3115	that there are cases where one side will attempt to re-establish communication when there is
3116	no persistent information for the relationship at the far-end. In such cases, it is important the
3117	side that is attempting recovery can distinguish between unsuccessful attempts to connect to
3118	the holder of the persistent information and when the information no longer exists. If the peer
3119	information does not exist, this side can draw conclusions and complete appropriately; if they
3120	merely fail to get through they are stuck in attempting recovery.
3121	
3122	Two mechanisms are provided to make it possible that even when one side of a
3122	Superior: Inferior relationship has completed, that a message can eventually get through to
3123	something that can definitively report the status, distinguishing this case from a temporary
3124	inability to access the state of a continuing transaction element. The mechanisms are:
	· ·
3126	o Address fields which provide a "callback address" can be a set of addresses,
3127	which are alternatives one of which is chosen as the target address for the
3128	future message. If the sender of that message finds the address does not work,
3129	it can try a different alternative.

3130 3131 3132 3133	o The REDIRECT message can be used to inform the peer that an address previously given is no longer valid and to supply a replacement address (or set of addresses). REDIRECT can be issued either as a response to receipt of a message or spontaneously.
3134 3135 3136	The two mechanisms can be used in combination, with one or more of the original set of
3137 3138	addresses just being a redirector, which does not itself ever have direct access to the state information for the transaction, but will respond to any message with an appropriate REDIRECT.
3139 3140 3141	An alternative implementation approach is to have a single addressable entity that uses the same address for all transactions, distinguishing them by identifier, and which always
3142 3143 3144	recovers to use the same address. Such an implementation would not need to supply "backup" addresses (and would only use REDIRECT if it was being permanently migrated).
3144 3145 3146	Terminator:Decider failures
3147 3148 3149	BTP does not provide facilities or impose requirements on the recovery of Terminator:Decider relationships, other than allowing messages to be repeated. A Terminator may survive failures (by retaining knowledge of the Decider's address and identifier), but this
3150 3151 3152 3153	is an implementation option. Although a Decider (if it decides to confirm) will persist information about the confirm decision, it is not required, after failure, to remain accessible using the inferior address it offered to the Terminator. Any such recovery is an implementation option.
3154 3155 3156	A Decider's address (as returned on BEGUN) may be a set of addresses, allowing a failed Decider to be recovered at a different address.
3157315831593160	A Decider has no way of initiating a call to a Terminator to ensure that it is still active, and thus no way of detecting that a Terminator has failed. To avoid a Decider waiting for ever for a CONFIRM_TRANSACTION that will never arrive, the standard qualifier "Transaction
3161 3162 3163 3164	timelimit" can be used (by the Initiator) to inform the Decider when it can assume the Terminator will not issue CONFIRM_TRANSACTION and so it (the Decider) should initiate cancellation.
3165	XML representation of Message Set
3166 3167 3168	This section describes the syntax for BTP messages in XML. These XML messages represent a midpoint between the abstract messages and what actually gets sent on the wire.
3169317031713172	All BTP related URIs have been created using Oasis URI conventions as specified in $\frac{\text{RFC}}{3121}$
3173 3174	The XML Namespace for the BTP messages is urn:oasis:names:tc:BTP:xml
3175 3176	In addition to an XML schema, this specification uses an informal syntax to describe the structure of the BTP messages. The syntax appears as an XML instance, but the values

3177 contain data types instead of values. The following symbols are appended to some of the
3178 XML constructs: ? (zero or one), * (zero or more), + (one or more.) The absence of one of
3179 these symbols corresponds to "one and only one."

3181 Addresses

As described in the "Abstract Message and Associated Contracts – Addresses" section, a BTP
address comprises three parts, and for a target address only the "additional information" field
is inside the BTP messages. For all BTP messages whose abstract form includes a target
address parameter, the corresponding XML representation includes a "target-additionalinformation" element. This element may be omitted if it would be empty.

For other addresses, all three fields are represent, as in:

3190
3191
3192
3193

3194

3195

3196 3197

3198 3199

3200

3201 3202

3203 3204

3205

3206

3207 3208 3209

3210 3211

3212 3213

3188 3189

3180

3182

<btp:some-address>
 <btp:binding-name>...carrier binding URI...</btp:binding-name>
 <btp:binding-address>...carrier specific
address...</btp:binding-address>
 <btp:additional-information>...optional additional addressing
information...</btp:additional-information> ?
</btp:some-address>

A "published" address can be a set of <some-address>, which are alternatives which can be chosen by the peer (sender.) Multiple addresses are used in two cases: different bindings to same endpoint, or backup endpoints. In the former, the receiver of the message has the choice of which address to use (depending on which binding is preferable.) In the case where multiple addresses are used for redundancy, a priority attribute can be specified to help the receiver choose among the addresses- the address with the highest priority should be used, other things being equal. The priority is used as a hint and does not enforce any behaviour in the receiver of the message. Default priority is a value of 1.

Qualifiers

The "Qualifier name" is used as the element name, within the namespace of the "Qualifier group".

Examples:

5215	Examples:
3214	<pre><btpq:inferior-timeout< pre=""></btpq:inferior-timeout<></pre>
3215	xmlns:btpq="urn:oasis:names:tc:BTP:qualifiers"
3216	xmlns:btp="urn:oasis:names:tc:BTP:xml"
3217	<pre>btp:must-be-understood="false"</pre>
3218	<pre>btp:to-be-propagated="false">1800</pre>
3219	
3220	<auth:username< td=""></auth:username<>
3221	<pre>xmlns:auth="http://www.example.com/ns/auth"</pre>
3222	xmlns:btp="urn:oasis:names:tc:BTP:xml"
3223	btp:must-be-understood="true"
3224	<pre>btp:to-be-propagated="true">jtauber</pre>
3225	

3226 3227	Attributes must-be-understood has default value "true" and to-be-propagated has default value "false".
3228	
3229	Identifiers
3230	
3230	Identifiers shall be UDIs Uppressified length strings made of up have desired disits $(0 > 0)$.
	Identifiers shall be URIs Unspecified length strings made of up hexadecimal digits (0>9, A-
3232	>F). Note: lower case a >f are not valid.
3233	
3234	Examples: "01", "FAB224234CCCC2"
3235	
3236	Note – Identifiers need to be unambiguous over all the systems that might be involved in a
3237	business transaction and over indefinite periods of time. Apart from their generation, Use of
3238	hexadecimal digits avoids problems with character code representations. Tthe only operation
3239	the BTP implementations have to perform on identifiers is to match them.
3240	
3241	Message References
	•
3242	Each BTP message has an optional id attribute to give it a unique identifier. An application
3243	can make use of those identifiers, but no processing is enforced.
3244	
3245	Messages
3246	
3247	CONTEXT
3247	CONTEXT
5240	
3249	<pre><htp:context id2="" id2_guperior_type="cohegion stom"></htp:context></pre>
3249 3250	<pre><btp:context id?="" superior-type="cohesion atom"> <btp:superior-address>_ +</btp:superior-address></btp:context></pre>
3250	<pre><btp:superior-address>- +</btp:superior-address></pre>
3250 3251	<pre><btp:superior-address>- + address</btp:superior-address></pre>
3250 3251 3252	<pre><btp:superior-address>- + address </btp:superior-address></pre>
3250 3251 3252 3253	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI</btp:superior-identifier></pre>
3250 3251 3252 3253 3254	<pre></pre>
3250 3251 3252 3253 3254 3255	<pre></pre>
3250 3251 3252 3253 3254	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address</btp:reply-address></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address </btp:reply-address></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3256 3257	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address</btp:reply-address></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ?</btp:qualifiers></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ? qualifiers</btp:qualifiers></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ? qualifiers </btp:qualifiers></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ? qualifiers </btp:qualifiers></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3263	
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ? qualifiers </btp:qualifiers></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266	<pre></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3265 3266 3267	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexstringURI <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ? qualifiers </btp:qualifiers> </btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexatringURIidentifier> <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ? qualifiers </btp:qualifiers> CONTEXT_REPLY <btp:context-reply id?="" superior-type="cohesion atom"> <btp:target-additional-information> ?</btp:target-additional-information></btp:context-reply></btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexatringURIidentifier> <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ? qualifiers </btp:qualifiers> CONTEXT_REPLY </btp:superior-identifier></pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269 3270	<pre> </pre>
3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269	<pre><btp:superior-address>- + address </btp:superior-address> <btp:superior-identifier>hexatringURIidentifier> <btp:reply-address> ? address </btp:reply-address> <btp:superior-type>cohesion atom</btp:superior-type> <btp:qualifiers> ? qualifiers </btp:qualifiers> CONTEXT_REPLY </btp:superior-identifier></pre>

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

Page 100 of 147

3274	<pre><btp:superior-identifier>hexstringURI</btp:superior-identifier></pre>
3275	identifier>
3276	<pre><btp:completion-< pre=""></btp:completion-<></pre>
3277	status>completed related repudiated
3278	<pre><btp:qualifiers> ?</btp:qualifiers></pre>
3279	qualifiers
3280	
3281	/btp:context-reply>
3282	
	UEST_STATUS
	UEST_STATUS
3284	
3285	<pre><btp:request-status id?=""></btp:request-status></pre>
3286	<pre></pre>
3287	additional address information
3288	<pre></pre>
3289	<pre></pre>
3290	address
3291	
3292	<pre><btp:target-identifier>URI</btp:target-identifier></pre>
3293	<pre></pre>
3294	qualifiers
3295	
3296	
3297	
3298 STA	TUS .
3299	
3300	
3301	<pre><btp:status id?=""></btp:status></pre>
	<pre></pre>
3302	additional address information
3303	<pre></pre>
3304	<pre></pre>
3305	
3306	<pre></pre>
3307	resigned preparing prepared
3308	confirming confirmed cancelling cancelled
3309	cancel-contradiction confirm-contradiction
3310	hazard contradicted unknown inaccessible
3311	value>
3312	<pre></pre>
3313	qualifiers
3314	
3315	
3316	
3317 FAU	IT
3318	
3319	<pre>chtn:foult_id0x</pre>
3320	<pre><btp:fault id?=""> </btp:fault></pre>
3320 3321	<pre></pre>
3322	additional address information
	<pre> </pre> <pre></pre> <pre>/// // // // // // // // // // // // //</pre>
3323	<pre></pre>
3324	<pre></pre>
3325	<pre></pre>

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

3326	<pre><btp:fault-data>fault data</btp:fault-data></pre> /btp:fault-data> ?
3327	<pre></pre>
3328	qualifiers
3329	
	<pre></pre>
3330	
3331	
3332	The following fault type names are represented by simple strings, corresponding to the entries
3333	defined in the abstract message set:
3334	
3335	o communication-failure
3336	o duplicate-inferior
3337	o general
3338	
3339	<u>o invalid-inferior</u>
3340	o invalid-superior
3341	o status-refused
3342	
3343	<u>o unknown-parameter</u>
3344	o unknown-transaction
3345	o unsupported-qualifier
3346	<u>o wrong-state</u>
	<u>o wrong-state</u>
3347	
3348	Revisions of this specification may add other fault type names, which shall be simple strings
3349	of letters, numbers and hyphens. If other specifications define fault type names to be used
3350	with BTP, the names shall be URIs.
3351	
3352	Fault data can take on various forms:
	<u>1 duit data can take on various forms.</u>
3353	
3354	Free text:
3355	
3356	<pre><btp:fault-data>string data</btp:fault-data></pre>
3357	
3358	Identifier:
3359	
3360	<pre><btp:fault-data>URI</btp:fault-data></pre>
	<pre></pre>
3361	
3362	
3363	Inferior Identity:
3364	
3365	<pre><btp:fault-data></btp:fault-data></pre>
3366	<pre></pre>
3367	address
3368	
3369	<pre></pre>
3370	<pre></pre>
	<pre>\/Dup+laulu-uala></pre>
3371	
3372	
3373	

3374 3375	BEGIN
3375	<pre><btp:begin id?="" transaction-type="cohesion atom"></btp:begin></pre>
3370	
3378	<pre></pre>
3378	
	<pre></pre>
3380	
3381	
3382	
3383	
3384	qualifiers
3385	
3386	
3387	
3388	
3389	BEGUN
3390	
3391	<pre><btp:begun id?="" transaction="" type="cohesion atom"></btp:begun></pre>
3392	<pre><btp:begun fu?="" transaction="" type="conesion[atom"> </btp:begun></pre>
3392	
3393	
	<pre></pre>
3395	
3396	address
3397	
3398	
3399	identifier> ?
3400	<pre></pre>
3401	
3402	address
3403	
3404	 btp:qualifiers>-?
3405	qualifiers
3406	
3407	
3408	
3409	
3410	ENROL
3411	
3412	<pre><btp:enrol id?="" reply-requested="true false"></btp:enrol></pre>
3413	<pre><btp:target-additional-information> ?</btp:target-additional-information></pre>
3414	additional address information
3415	
3415	<pre> <</pre>
3417	identifier>
3417	<pre>// identifier/ // stp:reply-requested/true false</pre> // btp:reply-requested/
3418	<pre></pre>
3419	
	address
3421	
3422	
3423	address
3424	

Page 103 of 147

```
3425
                  <btp:inferior-identifier>...hexstringURI.../btp:inferior-
3426
                identifier>
3427
                  <btp:qualifiers> ?
3428
                    ...qualifiers...
3429
                  </btp:qualifiers>
3430
                </btp:enrol>
3431
3432
3433
          ENROLLED
3434
3435
                <br/>dtp:enrolled id?>
3436
                <btp:target-additional-information> ?
3437
                    ...additional address information...
3438
                  </btp:target-additional-information>
3439
                  <btp:inferior-identifier>...hexstringURI.../btp:inferior-
3440
                identifier>
3441
                  <btp:inferior-handle>...hexstring.../btp:inferior:handle> ?
3442
                  <btp:qualifiers> ?
3443
                    ... qualifiers...
3444
                  </btp:qualifiers>
3445
                </btp:enrolled>
3446
3447
          RESIGN
3448
3449
3450
                <btp:resign response-requested="true|false"_id?>
3451
                <btp:target-additional-information>
3452
                    ...additional address information...
3453
                  </btp:target-additional-information>
3454
                  <btp:superior-identifier>...hexstringURI...</btp:superior-</pre>
3455
                identifier>
3456
                   <btp:inferior-address;</pre>
3457
                       .address.
3458
                  </btp:inferior-address>
3459
                  <btp:inferior-identifier>...hexstringURI...</btp:inferior-</pre>
3460
                identifier>
3461
                   <btp:response-requested>true | false</btp:response-requested>
3462
                   <btp:qualifiers> ?
3463
                    ...qualifiers...
3464
                  </btp:qualifiers>
3465
                </btp:resign>
3466
3467
          RESIGNED
3468
3469
3470
                <btp:resigned id?>
3471
                  <btp:target-additional-information> ?
3472
                    ...additional address information...
3473
                  </btp:target-additional-information>
3474
                  <btp:inferior-identifier>...hexstringURI...</btp:inferior-</pre>
3475
                identifier>
```

Page 104 of 147

```
3476
                                             <btp:qualifiers> ?
3477
                                                  ...qualifiers...
3478
                                             </btp:qualifiers>
3479
                                       </btp:resigned>
3480
3481
                         PREPARE
3482
3483
3484
                                       <br/>

3485
                                             <btp:target-additional-information> ?
3486
                                                  ...additional address information ...
3487
                                            </btp:target-additional-information>
3488
                                             <btp:inferior-identifier>...hexstringURI.../btp:inferior-
3489
                                       identifier>-?
3490
                                            <btp:qualifiers> ?
3491
                                                  ... qualifiers...
3492
                                             </btp:qualifiers>
3493
                                       </btp:prepare>
3494
3495
                         PREPARED
3496
3497
3498
                                       <btp:prepared default-is-cancel="false|true" id?>
3499
                                             <btp:target-additional-information> ?
3500
                                                  ...additional address information ...
3501
                                            </btp:target-additional-information>
3502
                                             <btp:superior-identifier>....hexstringURI....</btp:superior-</pre>
3503
                                       identifier>
3504
                                            3505
                                                         address.
3506
                                             </btp:inferior-address>
3507
                                            <btp:inferior-identifier>...hexstringURI.../btp:inferior-
3508
                                       identifier>
3509
                                             <btp:default-is-cancel>true|false</btp:default-is-cancel>
3510
                                            <btp:gualifiers> ?
3511
                                                  ... qualifiers...
3512
                                            </btp:qualifiers>
3513
                                       </btp:prepared>
3514
3515
3516
                         CONFIRM
3517
3518
                                       <br/>dtp:confirm id?>
3519
                                             <btp:target-additional-information> ?
3520
                                                  ...additional address information...
3521
                                            </btp:target-additional-information>
3522
                                             <btp:inferior-identifier>...hexstringURI...</btp:inferior-</pre>
3523
                                       identifier>
3524
                                             <btp:qualifiers> ?
3525
                                                  ... qualifiers...
3526
                                            </btp:qualifiers>
```

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

Page 105 of 147

```
3527
                </btp:confirm>
3528
3529
          CONFIRMED
3530
3531
                <btp:confirmed confirmed-received="true|false"_id?>
3532
3533
                  <btp:target-additional-information> ?
3534
                     ...additional address information ...
3535
                  </btp:target-additional-information>
3536
                  <btp:superior-identifier>...hexstringURI...</btp:superior-</pre>
3537
                identifier>
3538
                  <btp:inferior-address> ?
3539
                     ...address...
3540
                  </btp:inferior-address>
3541
                  <btp:inferior-identifier>....hexstringURI....</btp:inferior-</pre>
3542
                identifier>-?
                  <btp:confirmed-received>true|false</btp:confirmed-received>
3543
3544
                  <btp:qualifiers> ?
3545
                    ... qualifiers...
3546
                  </btp:qualifiers>
3547
                </btp:confirmed>
3548
3549
          CANCEL
3550
3551
3552
                <br/>dtp:cancel id?>
3553
                  <btp:target-additional-information> ?
                     ...additional address information ...
3554
3555
                  </btp:target-additional-information>
3556
                  <btp:inferior-identifier>...hexstringURI...</btp:inferior-</pre>
3557
                identifier>-?
3558
                  <btp:reply-address> ?
3559
                    ...address...
3560
                  </btp:reply-address>
3561
                   <btp:qualifiers> ?
3562
                    ... qualifiers...
                  </btp:qualifiers>
3563
3564
                </btp:cancel>
3565
3566
          CANCELLED
3567
3568
3569
                <br/>dtp:cancelled id?>
3570
                  <btp:target-additional-information> ?
3571
                    ...additional address information...
3572
                  </btp:target-additional-information>
3573
                  <btp:superior-identifier>...hexstringURI...</btp:superior-</pre>
3574
                identifier>
3575
                  <btp:inferior-address>
3576
                      ...address...
                   </btp:inferior-address> ?
3577
```

Page 106 of 147

3578	<pre><btp:inferior-identifier>hexstringURI</btp:inferior-identifier></pre>	
3579	identifier> ?	
3580	<pre></pre>	
3581	qualifiers	
3582	/btp:qualifiers>	
3583		
3584		
3585		
	ONFIRM_ONE_PHASE	
3587		
3588	<pre><btp:confirm-one-phase _id?="" report-hazard="true false"></btp:confirm-one-phase></pre>	
3589	<pre><btp:target-additional-information> ?</btp:target-additional-information></pre>	
3590	additional address information	
3591		
		1
3592	<pre></pre>	
3593	identifier>	
3594	<pre><btp:report-hazard>true false</btp:report-hazard></pre>	
3595	<pre><btp:qualifiers> ?</btp:qualifiers></pre>	
3596	qualifiers	
3597		
3598		
	() beprediffing one phase?	
3599		
3600 HA	AZARD	
3601		
3602	<pre><btp:hazard _id?="" level="mixed possible"></btp:hazard></pre>	
3603		
	<pre><btp:target-additional-information>_?</btp:target-additional-information></pre>	I
3604	additional address information	
3605		
3606	<pre><btp:superior-identifier>hexstringURI</btp:superior-identifier></pre>	
3607	identifier>	
3608		
3609		
3610		
3611		
	<pre></pre>	
3612	identifier>	
3613	<pre><btp:level>mixed possible</btp:level></pre>	
3614	<pre><btp:qualifiers> ?</btp:qualifiers></pre>	
3615	qualifiers	
3616		
3617		
3618		
3619		
3620 CC	DNTRADICTION	
3621		
3622	abte: contradiction id?	
	<pre><btp:contradiction id?=""></btp:contradiction></pre>	1
3623	<pre><btp:target-additional-information> ?</btp:target-additional-information></pre>	
3624	additional address information	
3625		
3626	<pre><btp:inferior-identifier>hexstringURI</btp:inferior-identifier></pre> /btp:inferior-	
3627	identifier>	I
3628	<pre><btp:qualifiers> ?</btp:qualifiers></pre>	

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

Page 107 of 147

2620		
3629	qualifiers	
3630		
3631		
3632		
3633		
3634	SUPERIOR_STATE	
3635	SOI ENOR_STATE	
		1
3636	<pre><btp:superior-state _id?="" reply-requested="true false"></btp:superior-state></pre>	
3637	<pre><btp:target-additional-information>_?</btp:target-additional-information></pre>	I
3638	additional address information	
3639		1
3640	<pre></pre>	I
3641	identifier>	
3642	<pre><btp:status>active prepared-</btp:status></pre>	
3643	received inaccessible unknown	
3644	<pre></pre>	I
3645	<pre><btp:qualifiers> ?</btp:qualifiers></pre>	
3646	qualifiers	
3647		
3648		
3649		
3650		
3651	INFERIOR_STATE	
3652		
3653	<pre><btp:inferior-state _id?="" reply-requested="true false"></btp:inferior-state></pre>	
3654	<pre><btp:filefior=state id;="" raise="" repry="" requested="true"> <btp:target-additional-information> ?</btp:target-additional-information></btp:filefior=state></pre>	
3655	additional address information	I
3656		
3657		1
	<pre></pre>	
3658		1
3658	identifier>	1 1
3659		
3659 3660		
3659 3660 3661		
3659 3660 3661 3662	<pre></pre>	
3659 3660 3661 3662 3663	<pre></pre>	
3659 3660 3661 3662 3663 3664	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3666 3667	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3666 3667 3668	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3673	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3673 3674 3675	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3673 3674 3675 3676	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674 3675 3676 3677	<pre></pre>	
3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3673 3674 3675 3676	<pre></pre>	

3680	<pre><btp:superior-identifier>hexstringURI</btp:superior-identifier></pre>	
3681	identifier> ?	1
3682	<pre></pre>	Í
3683	identifier>	
3684	<pre> <btp:old-address> +</btp:old-address></pre>	
3685	address	
3686		
3687	<pre></pre>	
3688	address	
3689		
3690	<pre> <btp:qualifiers> ?</btp:qualifiers></pre>	
3691	qualifiers	
3692		
3693		
3694		
		I
3695 <u>BEC</u>	<u>NIc</u>	
3696		
3697	 btp:begin id?>	
3698	<pre> <</pre>	
3699		
	additional address information	
3700	<pre></pre>	
3701	<pre></pre>	
3702	address	
3703		
3704	<pre><btp:transaction-type>cohesion atom</btp:transaction-type></pre>	
3705	<pre></pre>	
3706		
	qualifiers	
3707	<pre>/btp:qualifiers></pre>	
3708		
3709		
3710		
3711 <u>BEC</u>		
3712		
3713	<pre></pre>	
3714	<pre></pre>	
3715	additional address information	
3716	<pre> </pre>	
3717		
	<pre></pre>	
3718	address	
3719		
3720	<pre></pre>	
3721	identifier> ?	
3722	<pre></pre>	
3723	<pre></pre>	
3724	address	
3724		
	<pre></pre>	
3726	<pre></pre>	
3727	qualifiers	
3728		
3729		
3730		
3731		

Page 109 of 147

3732 PREPARE_INFERIORS3733

3734	<pre><btp:-prepare-inferiors id?=""></btp:-prepare-inferiors></pre>
3735	<pre><btp:target-additional-information> ?</btp:target-additional-information></pre>
3736	additional address information
3737	
3738	<pre></pre>
3739	address
3740	
3741	<pre></pre>
3742	identifier>-?
3743	<pre> <</pre>
3744	<pre></pre>
	handle> +
3746	
3747	<pre><btp:qualifiers> ?</btp:qualifiers></pre>
3748	qualifiers
3749	/btp:qualifiers>
3750	
3751	() bep · prepare inferiors/
3752	
	RM_TRANSACTION
3754	
3755	<pre><btp:confirm-transaction _id?="" report-hazard="true false"></btp:confirm-transaction></pre>
3756	<pre><btp:target-additional-information> ?</btp:target-additional-information></pre>
3757	additional address information
3758	
3759	<pre><btp:reply-address> ?</btp:reply-address></pre>
3760	address
3761	
3762	<pre></pre>
3763	identifier>
3764	<pre><btp:inferiors-list> ?</btp:inferiors-list></pre>
3765	<pre><btp:inferior-handle>hexstringURI</btp:inferior-handle></pre>
3766	handle> +
3767	
3768	<pre><btp:report-hazard>true false</btp:report-hazard></pre>
3769	<pre><btp:qualifiers> ?</btp:qualifiers></pre>
3770	qualifiers
3771	
3772	
3773	
3774	
3775 TRANS	SACTION_CONFIRMED
3776	
3777	<pre><btp:transaction-confirmed id?=""></btp:transaction-confirmed></pre>
3778	<pre><btp:transaction-confirmed id;=""> <btp:target-additional-information> ?</btp:target-additional-information></btp:transaction-confirmed></pre>
3779	additional address information
3780	<pre>/btp:target-additional-information></pre>
3781	
3782	
3783	
5105	

3784	<pre><btp:transaction-identifier>hexstringURI</btp:transaction-identifier></pre>
3785	identifier>
3786	<pre><btp:qualifiers> ?</btp:qualifiers></pre>
3787	qualifiers
3788	
3789	
3790	
3791	
3792	CANCEL_TRANSACTION
3793	
3793	<pre><btp:canceltransaction id?=""></btp:canceltransaction></pre>
3795	<pre> <</pre>
3796	additional address information
3790	<pre></pre>
3798	
3798	<pre><btp:reply-address> -?</btp:reply-address></pre>
3800	address
3800	<pre></pre>
3801	<pre><btp:transaction-identifier>hexstringURIidentifier>?</btp:transaction-identifier></pre>
3802	
3803	<pre></pre>
3804	<pre></pre>
3805	qualifiers
3800	//btp:qualifiers>
3808	
3809	CANCEL_INFERIORS
3810	
3811	<pre><btp:cancel-inferiors id?=""></btp:cancel-inferiors></pre>
3812	<pre><btp:target-additional-information> ?</btp:target-additional-information></pre>
3813	additional address information
3814	
3815	<pre><btp:reply-address>- ?</btp:reply-address></pre>
3816	address
3817	
3818	<pre></pre>
3819	identifier> ?
3820	<pre><btp:inferiors-list></btp:inferiors-list></pre>
3821	<pre></pre>
3822	handle>_+
3823	
3824	<pre></pre>
3825	qualifiers
3826	
3827	
3828	
3829	
3830	TRANSACTION_CANCELLED
3831	
3832	<pre><btp:cancel-completetransaction-cancelled id?=""></btp:cancel-completetransaction-cancelled></pre>
3833	<pre><btp:target-additional-information>_?</btp:target-additional-information></pre>
3834	additional address information

Page 111 of 147

3835	
3836	
3837	address
3838	
3839	<pre><btp:transaction-identifier>hexstringURI</btp:transaction-identifier></pre>
3840	identifier>-?
3841	<pre></pre>
3842	qualifiers
3843	
3844	
3845	
3846	
	IEST_INFERIOR_STATUSES
	EST_INFERIOR_STATUSES
3848	
3849	<pre><btp:request_inferiorstatuses id?=""></btp:request_inferiorstatuses></pre>
3850	<pre> <btp:target-additional-information> ?</btp:target-additional-information></pre>
3851	additional address information
3852	
3853	<pre></pre>
3854	address
3855	
3856	<pre></pre> /btp:target-identifier>hexstringURI/btp:target-
3857	identifier>
3858	<pre></pre>
3859	<pre><btp:inferior-handle>hexstringURI</btp:inferior-handle></pre>
3860	handle> +
3861	
3862	<pre></pre>
3863	qualifiers
3864	
3865	
3866	
3867	
3868 INFER	RIOR_STATUSES
3869	
3870	<btp:inferiorstatuses id?=""></btp:inferiorstatuses>
3871	<pre> <btp:target-additional-information> ?</btp:target-additional-information></pre>
3872	additional address information
3873	
3874	<pre></pre>
3875	
3876	
3877	<pre></pre>
3878	identifier>
3879	<btp:status-list></btp:status-list>
3880	<pre><btp:status-item> +</btp:status-item></pre>
3881	<pre><btp:inferior-handle>hexstringURI</btp:inferior-handle></pre>
3882	handle>
3883	<pre><btp:status>active resigned preparing prepared </btp:status></pre>
3884	autonomously-confirmed autonomously-cancelled
3885	confirming confirmed cancelling cancelled
3886	cancel-contradiction confirm-contradiction
2000	

Page 112 of 147

3887	hazard invalid
3888	<pre></pre>
3889	qualifiers
3890	/btp:qualifiers>
3891	
3892	
3893	<pre> <</pre>
3894	qualifiers
3895	
3896	
3897	
3898	
3899	REQUEST_STATUS
3900	
3901	<pre><btp:request_status_id?></btp:request_status_id?></pre>
3902	
3903	
3904	<pre></pre>
3905	
3906	
3907	
3908	<pre></pre>
3909	<pre></pre>
3910	
3911	
3912	
	() Dep · requebe_beacaby
3013	
3913 2014	STATUS
3914	STATUS
3914 3915	
3914 3915 3916	<pre><btp:status id?=""></btp:status></pre>
3914 3915 3916 3917	<pre><btp:status_id?><btp:target-additional_information></btp:target-additional_information></btp:status_id?></pre>
3914 3915 3916 3917 3918	<pre><btp:status id?=""> </btp:status></pre>
3914 3915 3916 3917 3918 3919	<pre><btp:status_id?> </btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920	<pre><btp:status id?=""> </btp:status></pre>
3914 3915 3916 3917 3918 3919 3920 3921	<pre><btp:status id?=""></btp:status></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923	 <btp:status_id?> <btp:target-additional-information> additional_address_information </btp:target-additional-information> <btp:responder-address> address </btp:responder-address> <btp:responder-identifier>hexstring</btp:responder-identifier></btp:status_id?>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3929	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3926 3927 3928 3929 3930	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3927 3928 3929 3930 3931	<pre><btp:status-id?></btp:status-id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3929 3930 3931 3932	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3927 3928 3929 3930 3931	<pre></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3929 3930 3931 3932	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3929 3930 3931 3932 3933	<pre></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3929 3930 3931 3932 3933 3934	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3929 3930 3931 3932 3931 3932 3933 3934 3935 3936	<pre><btp:status_id?></btp:status_id?></pre>
3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3927 3928 3929 3930 3931 3932 3933 3934 3935	<pre><btp:status_id?></btp:status_id?></pre>

Page 113 of 147

3939	<pre><btp:fault_id?></btp:fault_id?></pre>
3940	<pre></pre>
3941	
3942	
3943	
3944	identifier> ?
3945	
3946	identifier> ?
3947	<pre></pre>
3948	
3949	
3950	
3951	
3952	
3953	
3954	
3955	The following fault type names are represented by simple strings, corresponding to the entries
3956	defined in the abstract message set:
3957	
3958	ogeneral
3959	ounknown-parameter
3960	owrong-state
	ocommunication failure
3961	
3962	oinvalid superior
3963	oduplicate-inferior
3964	ounknown-inferior
3965	
3966	Revisions of this specification may add other fault type names, which shall be simple strings
3967	of letters, numbers and hyphens. If other specifications define fault type names to be used
3968	with BTP, the names shall be URIs.
	with DTF, the fidnes shall be URIS.
3969	
3970	Fault data can take on various forms:
3971	
3972	Free text:
3973	
3974	<pre><btp:fault-data>string data</btp:fault-data></pre>
3975	
3976	Identifier:
	iuchtinei.
3977	
3978	<pre><btp:fault-data>hexstring</btp:fault-data></pre> /btp:fault-data>
3979	—
3980	
3981	Inferior Identity:
3982	
3983	<pre><btp:fault_data></btp:fault_data></pre>
3984	
3985	
3986	
3987	
3988	identifier>
5700	

Page 114 of 147

The inf	rd qualifiers formal syntax for these messages assumes the namespace prefix "btpq" is associate URI "urn:oasis:names:tc:BTP:qualifiers".
Transa	action timelimit
	<btpq:transaction-timelimit></btpq:transaction-timelimit>
	<pre><btpq:timelimit> time_in_reserved</btpq:timelimit></pre>
	time in seconds
	transaction-timelimit>
Inferio	r timeout
	<btpq:inferior-timeout></btpq:inferior-timeout>
	<pre><btpq:timeout></btpq:timeout></pre>
	time in seconds
	 <btpq:intended-decision>confirm cancel</btpq:intended-decision>
Minim	um inferior timeout
	<pre><btpq:minimum-inferior-timeout></btpq:minimum-inferior-timeout></pre>
	<btpq:minimum-timeout></btpq:minimum-timeout>
	time in seconds
Inferio	r name
mene	<pre><btpg:inferior-name></btpg:inferior-name></pre>
	<pre><btpq:inferior-name></btpq:inferior-name></pre>
	<pre></pre>
	-/ PcFd. THIET TOT_Hame>
Compo	unding of Messages
Relatin	g BTP to one another, in a "group" is represented by containing them within the
	ated_group element, with the related messages as child elements. The processing
	up is defined in the section "Groups – combinations of related messages". For e
	(htp://wolated.group)
	<pre><btp:related_group></btp:related_group></pre>
	<completion-status>related</completion-status>
	<btp:enrol></btp:enrol>
	<pre></pre>

Page 115 of 147

4038	
4039 4040	If the rules for the group state that the target address of the abstract message is omitted, the
4040	corresponding target-address-information element shall be absent in the message in the
4042	related_group. The carrier protocol binding specifies how a relation between application and
4043 4044	BTP messages is represented.
4044	Bundling (semantically insignificant combination) of BTP messages and related groups is
4046	indicated with the "btp:messages" element, with the bundled messages and related groups as
4047 4048	child elements. For example (confirming one and cancelling another inferiors of a cohesion):
4048	<pre>>btp:messages></pre>
4050	<pre><btp:confirm></btp:confirm></pre>
4051 4052	<pre></pre>
4053	
4054	
4055	

	XML Schemas
	XML schema for BTP messages
_	<pre><?xml version="1.0"?></pre>
-	< <u>schema</u>
-	<pre>xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:oasis:names:tc:BTP:xml"</pre>
-	xmlns:btp="urn:oasis:names:tc:BTP:xml"
	<pre>elementFormDefault="qualified"></pre>
-	Qualifiers
	<complextype name="qualifier-type"></complextype>
	<pre><simplecontent></simplecontent></pre>
	<pre><extension base="string"></extension></pre>
	<attribute name="must-be-understood" type="boolean"></attribute>
	<attribute name="to-be-propagated" type="boolean"></attribute>
_	cvtension
-	<pre></pre>
	<pre><element <="" abstract="true" name="qualifier" pre="" type="btp:qualifier-type"></element></pre>
	<pre><element name="qualifiers"></element></pre>
-	<pre><complextype></complextype></pre>
	<pre></pre>
	<pre><element maxoccurs="unbounded" ref="btp:qualifier"></element></pre>
_	
_	
-	
	example qualifier:</td
	<pre><!-- example qualifier. <element name="some-qualifer" type="btp:qualifier-type"</pre--></pre>
5	substitutionGroup="btp:qualifier"/>
-	>
-	Message set data types
-	<pre><simpletype name="identifier"> <restriction base="anyURI"></restriction></simpletype></pre>
-	<pre> </pre>
-	
	<pre><simpletype name="additional-information"></simpletype></pre>
	<pre><restriction base="string"></restriction></pre>
-	<complextype name="address"></complextype>
_	<sequence></sequence>

4107	colement neme-"binding neme" type-"envIIDI" />
4107 4108	<pre><element name="binding-name" type="anyURI"></element> <element name="binding-address" type="string"></element></pre>
4108	<pre><element minoccurs="0" name="additional-information" type="btp:additional-</pre></th></tr><tr><th>4110</th><th>information"></element></pre>
4111	<pre>//sequence></pre>
4112	
4113	
4114	<pre><simpletype name="superior-type"></simpletype></pre>
4115	<pre><restriction base="string"></restriction></pre>
4116	<pre><enumeration value="cohesion"></enumeration></pre>
4117	<pre><enumeration value="atom"></enumeration></pre>
4118	
4119	
4120	
4121	<pre><simpletype name="transaction-type"></simpletype></pre>
4122	<restriction base="string"></restriction>
4123	<pre><enumeration value="cohesion"></enumeration></pre>
4124	<pre><enumeration value="atom"></enumeration></pre>
4125	<pre></pre>
4126	
4127	
4128 4129	
4129	<pre><!-- Compounding--></pre>
4130	<pre><element name="messages"></element></pre>
4132	<pre><complextype></complextype></pre>
4132	<pre> <sequence></sequence></pre>
4134	<pre><element <="" minoccurs="0" pre="" ref="btp:message"></element></pre>
4135	maxOccurs="unbounded"/>
4136	
4137	
4138	
4139	
4140	<pre><element name="related-group" substitutiongroup="btp:message"></element></pre>
4141	<pre><complextype></complextype></pre>
4142	<pre><sequence></sequence></pre>
4143	<pre><element <="" minoccurs="0" pre="" ref="btp:message"></element></pre>
4144	maxOccurs="unbounded"/>
4145	<pre></pre>
4146	<pre></pre>
4147 4148	
4148	
4149	Message set
4151	C: Message Set
4152	<pre><element abstract="true" name="message"></element></pre>
4153	
4154	<pre><element name="context" substitutiongroup="btp:message"></element></pre>
4155	<pre><complextype></complextype></pre>
4156	<pre><sequence></sequence></pre>
4157	<pre><element <="" name="superior-address" pre="" type="btp:address"></element></pre>
4158	maxOccurs="unbounded"/>
4159	<pre><element name="superior-identifier" type="btp:identifier"></element></pre>

	<pre><element btp:qualifiers"="" minoccurs="0" name="superior-type" type="btp:superior-type</pre></th></tr><tr><th></th><th><pre><element ref="></element></pre>
	<attribute name="id" type="ID" use="optional"></attribute>
	<pre><element name="context-reply" substitutiongroup="btp:message"></element></pre>
	<pre><complextype></complextype></pre>
	<pre><sequence></sequence></pre>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
ΞY	pe="btp:additional-information" minOccurs="0"/>
	<pre><element completion-status"="" name="superior-identifier" type="btp:identif</pre></td></tr><tr><td></td><td><pre><element name="></element></pre>
	<pre><simpletype></simpletype></pre>
	<restriction base="string"></restriction>
	<enumeration value="completed"></enumeration>
	<pre><enumeration value="related"></enumeration></pre>
	<pre><enumeration value="repudiated"></enumeration></pre>
	<element minoccurs="0" ref="btp:qualifiers"></element>
	<attribute name="id" type="ID"></attribute>
	<pre><element name="request-status" substitutiongroup="btp:message"></element></pre>
	<complextype></complextype>
	<sequence></sequence>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
ty	<pre>pe="btp:additional-information" minOccurs="0"/> </pre>
	<pre><element <="" name="reply-address" pre="" type="btp:address"></element></pre>
mı	nOccurs="0"/>
	<pre><element <="" name="target-identifier" pre="" type="btp:identifie"></element></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<pre></pre>
	<attribute name="id" type="ID"></attribute>
	<pre></pre>
	colomont nemo-"status" substitution Guarda "http://www.status.com
	<pre><element name="status" substitutiongroup="btp:message"></element></pre>
	<pre><complextype></complextype></pre>
	<pre><sequence></sequence></pre>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
+	
ty	pe="btp:additional-information" minOccurs="0"/>
	<pre><element <="" name="responders-identifier" pre=""></element></pre>

<pre></pre> <pre><</pre>
<pre> </pre> <pre> <enumeration value="enrolling"></enumeration> <enumeration value="active"></enumeration> <enumeration value="active"></enumeration> <enumeration value="resigning"></enumeration> <enumeration value="resigned"></enumeration> <enumeration value="prepared"></enumeration> <enumeration value="prepared"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradicted"></enumeration> <enumeration value="confirm-contradicted"></enumeration> <enumeration value="confirm-contradicted"></enumeration> <enumeration value="confirm-contradicted"></enumeration> </pre>
<pre> <pre> <pre> <enumeration value="active"></enumeration> <enumeration value="resigning"></enumeration> <enumeration value="resigning"></enumeration> <enumeration value="prepared"></enumeration> <enumeration value="prepared"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="concelled"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration td="" va<=""></enumeration></pre></pre></pre>
<pre> <enumeration value="resigning"></enumeration> <enumeration value="resigned"></enumeration> <enumeration value="resigned"></enumeration> <enumeration value="preparing"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="cancelling"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="cancel-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="unknown"></enumeration> <enumeration value="inaccessible"></enumeration> </pre>
<pre> <enumeration value="resigned"></enumeration> <enumeration value="preparing"></enumeration> <enumeration value="prepared"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="cancelling"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="unknown"></enumeration> <enumeration value="inaccessible"></enumeration> <!--/restriction--> > >>>>>>>>></pre>
<pre> <enumeration value="preparing"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirming"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="confirmed"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confirm-contradicted"></enumeration> <enumeration value="unknown"></enumeration> <enumeration value="inaccessible"></enumeration> </pre>
<pre></pre>
<pre></pre>
<pre><enumeration value="confirmed"></enumeration> <enumeration value="cancelling"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="cancel-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="confradicted"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="inaccessible"></enumeration> </pre>
<pre> <enumeration value="cancelling"></enumeration> <enumeration value="cancelled"></enumeration> <enumeration value="cancel-contradiction"></enumeration> <enumeration value="cancel-contradiction"></enumeration> <enumeration value="confirm-contradiction"></enumeration> <enumeration value="hazard"></enumeration> <enumeration value="hazard"></enumeration> <enumeration value="contradicted"></enumeration> <enumeration value="unknown"></enumeration> <enumeration value="inaccessible"></enumeration> <enumeration value="inaccessible"></enumeration> </pre>
<pre></pre>
<pre><pre></pre></pre>
<pre><enumeration value="confirm-contradiction"></enumeration></pre>
<pre></pre>
<pre></pre> <
<pre></pre> <pre><</pre>
<pre></pre>
<pre></pre>
<pre></pre>
<pre> <attribute name="id" type="ID"></attribute> <complextype> <sequence> <element name="fault" substitutiongroup="btp:message"> </element> </element> </element> </element> </element> </sequence></complextype></pre>
<pre> <attribute name="id" type="ID"></attribute> <element name="fault" substitutiongroup="btp:message"> <complextype> <complextype> <element cype="btp:additional-information" minoccurs="0" name="target-additional-information"></element> </complextype></complextype></element></pre>
<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
<pre> <attribute name="id" type="ID"></attribute> <complextype> <sequence> <sequence> <element <="" name="target-additional-information" td=""></element></sequence></sequence></complextype></pre>
<pre><attribute name="id" type="ID"></attribute> </pre>
<pre></pre>
<pre> </pre>
<pre><element name="fault" substitutiongroup="btp:message"></element></pre>
<pre><complextype> <sequence> <element minoccurs="0" name="target-additional-information" type="btp:additional-information"></element></sequence></complextype></pre>
<pre><complextype> </complextype></pre> <pre><complextype> </complextype></pre> <pre><clement <="" name="target-additional-information" pre=""> <pre>type="btp:additional-information" minOccurs="0"/> </pre></clement></pre>
<pre><sequence> <element <="" name="target-additional-information" td=""></element></sequence></pre>
<pre><element minoccurs="0" name="target-additional-information" xype="btp:additional-information"></element></pre>
<pre>cype="btp:additional-information" minOccurs="0"/></pre>
ninOccurs="0"/>
<pre><element 0"="" name="inferior-identifier" type="btp:identif.</pre></td></tr><tr><td>ninOccurs="></element></pre>
<pre><element name="fault-type"></element></pre>
<pre></pre>
<pre><restriction base="string"></restriction></pre>
<pre><enumeration value="communication-failure"></enumeration> </pre>
<pre><enumeration value="duplicate-inferior"></enumeration></pre>
<pre></pre>
<pre></pre>
<pre><enumeration value="invalid-inferior"></enumeration></pre>
<pre><enumeration value="invalid-superior"></enumeration></pre>
<pre><enumeration value="status-refused"></enumeration></pre>
<pre><enumeration value="invalid-terminator"></enumeration></pre>
<pre><enumeration value="unknown-parameter"></enumeration></pre>
<pre><enumeration value="unknown-transaction"></enumeration></pre>
<enumeration value="unsupported-qualifier"></enumeration>
<pre><enumeration value="wrong-state"></enumeration></pre>

6	
7	<pre><element minoccurs="0" name="fault-data" type="anyType"></element></pre>
8	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<pre></pre>
	<attribute name="id" type="ID"></attribute>
	<pre><element name="enrol" substitutiongroup="btp:message"></element></pre>
	<complextype></complextype>
	<sequence></sequence>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
1	<pre>type="btp:additional-information" minOccurs="0"/></pre>
_	<pre><element name="superior-identifier" type="btp:identifier"></element></pre>
_	<pre><element name="reply-requested" type="boolean"></element></pre>
_	<pre><element <="" name="reply-address" pre="" type="btp:address"></element></pre>
r	ninOccurs="0"/>
_	<pre><element <="" name="inferior-address" pre="" type="btp:address"></element></pre>
r	ninOccurs="1" maxOccurs="unbounded"/>
_	<pre><element name="inferior-identifier" type="btp:identifier"></element></pre>
_	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<attribute name="id" type="ID"></attribute>
	<pre><element name="enrolled" substitutiongroup="btp:message"></element></pre>
	<sequence></sequence>
_	<pre><element <="" name="target-additional-information" pre=""></element></pre>
t	<pre>cype="btp:additional-information" minOccurs="0"/></pre>
_	<pre><element name="inferior-identifier" type="btp:identifier"></element></pre>
_	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
-	<attribute name="id" type="ID"></attribute>
_	
	<pre><element name="resign" substitutiongroup="btp:message"></element></pre>
	<complextype></complextype>
	<pre><sequence></sequence></pre>
-	<pre><element <="" name="target-additional-information" pre=""></element></pre>
1	<pre>type="btp:additional-information" minOccurs="0"/> type="btp:additional-information" minOccurs="0"/> type="btp:addition" minOccurs="0"/>> type="btp:addition" minOccurs="0"/>> type="btp:addition" minOccurs</pre>
_	<pre><element name="superior-identifier" type="btp:identifier"></element> </pre>
-	<pre><element name="inferior-identifier" type="btp:identifier"></element> <element name="response-requested" type="boolean"></element></pre>
_	<pre><element name="response-requested" type="boolean"></element> <element minoccurs="0" ref="btp:qualifiers"></element></pre>
-	<pre><element minoccurs="0" rel="btp:qualifiers"></element> </pre>
	<pre> <attribute name="id" type="ID"></attribute></pre>
Ī	<pre></pre>

9	<pre><element name="resigned" substitutiongroup="btp:message"></element></pre>
0	<pre><complextype></complextype></pre>
1	<pre><sequence></sequence></pre>
2	<pre></pre>
3	type="btp:additional-information" minOccurs="0"/>
4	<pre><element name="inferior-identifier" type="btp:identifier"></element></pre>
5	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
6	
7	<attribute name="id" type="ID"></attribute>
8	
9	
)	
	<pre><element name="prepare" substitutiongroup="btp:message"></element></pre>
	<complextype></complextype>
	<sequence></sequence>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
	<pre>type="btp:additional-information" minOccurs="0"/></pre>
	<pre><element name="inferior-identifier" type="btp:identifier"></element></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<attribute name="id" type="ID"></attribute>
	<pre><element name="prepared" substitutiongroup="btp:message"></element></pre>
	<complextype></complextype>
	<pre><sequence></sequence></pre>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
	type="btp:additional-information" minOccurs="0"/>
	<pre><element name="superior-identifier" type="btp:identifier"></element></pre>
	<pre><element name="inferior-identifier" type="btp:identifier"></element></pre>
	<pre><element name="default-is-cancel" type="boolean"></element></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<attribute name="id" type="ID"></attribute>
	<pre><element name="confirm" substitutiongroup="btp:message"></element></pre>
	<pre><complextype></complextype></pre>
	<sequence></sequence>
	<pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre></pre>
	type="btp:additional-information" minOccurs="0"/>
	<pre><element name="inferior-identifier" type="btp:identifier"></element></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<attribute name="id" type="ID"></attribute>
	<pre><element name="confirmed" substitutiongroup="btp:message"></element></pre>
	<pre><complextype></complextype></pre>
	<pre><sequence></sequence></pre>

	<pre><element <="" name="target-additional-information" pre=""></element></pre>
t	<pre>ype="btp:additional-information" minOccurs="0"/></pre>
	<pre><element <="" name="superior-identifier" pre="" type="btp:identifier"></element></pre>
	<pre><element <="" name="inferior-identifier" pre="" type="btp:identifier"></element></pre>
	<pre><element name="confirmed-received" type="boolean"></element></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
_	<pre></pre>
	<pre><attribute name="id" type="ID"></attribute></pre>
_	<pre></pre>
	<pre><element name="cancel" substitutiongroup="btp:message"></element></pre>
_	<pre></pre>
_	_
_	<pre></pre>
_	<pre>ype="btp:additional-information" minOccurs="0"/></pre>
	<pre><element <="" name="inferior-identifier" pre="" type="btp:identifier"></element></pre>
	<pre><element <="" name="reply-address" pre="" type="btp:address"></element></pre>
n	inOccurs="0"/>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<attribute name="id" type="ID"></attribute>
	<pre><element name="cancelled" substitutiongroup="btp:message"></element></pre>
	<complextype></complextype>
	<pre><sequence></sequence></pre>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
E	<pre>ype="btp:additional-information" minOccurs="0"/></pre>
	<pre><element <="" name="superior-identifier" pre="" type="btp:identifier"></element></pre>
	<pre><element <="" name="inferior-identifier" pre="" type="btp:identifier"></element></pre>
m	inOccurs="0"/>
-	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<pre></pre>
	<pre><attribute name="id" type="ID"></attribute></pre>
	<pre></pre>
	<pre><element name="confirm-one-phase" substitutiongroup="btp:message"></element></pre>
	<pre>complexType></pre>
	<pre><sequence></sequence></pre>
_	<pre></pre>
t	<pre>ype="btp:additional-information" minOccurs="0"/></pre>
	<pre><element name="inferior-identifier" report-hazard"="" type="boolean"></element></pre>
	<element minoccurs="0" ref="btp:qualifiers"></element>
	<pre><attribute name="id" type="ID"></attribute></pre>
ſ	

Page 123 of 147

	<complextype></complextype>
	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
	ype="btp:additional-information" minOccurs="0"/>
	<pre><element <="" inferior-identifier"="" name="superior-identifier" pre="" type="btp:identifier"></element></pre>
	<pre><element name="level"></element></pre>
ĺ	<simpletype></simpletype>
	<restriction base="string"></restriction>
	<pre><enumeration value="mixed"></enumeration></pre>
	<pre><enumeration value="possible"></enumeration></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
J	<attribute name="id" type="ID"></attribute>
	<pre><element name="contradiction" substitutiongroup="btp:message"></element></pre>
	<complextype></complextype>
	<pre><sequence></sequence></pre>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
-	<pre>ype="btp:additional-information" minOccurs="0"/></pre>
	<pre><element btp:qualifiers"="" minoccurs="0" name="inferior-identifier" type="btp:identifier</pre></td></tr><tr><td>_</td><td><element ref="></element></pre>
_	
	<attribute name="id" type="ID"></attribute>
_	
_	
	<pre><element name="superior-state" substitutiongroup="btp:message"></element></pre>
_	<pre><complextype></complextype></pre>
	<pre><sequence></sequence></pre>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
t	<pre>ype="btp:additional-information" minOccurs="0"/></pre>
	<pre><element <="" name="inferior-identifier" pre="" type="btp:identifier"></element></pre>
	<pre><element name="status"></element></pre>
	<pre><simpletype></simpletype></pre>
_	<restriction base="string"></restriction>
_	<pre>conumeration_walue="active"/></pre>
	<pre><enumeration value="active"></enumeration></pre>
	<pre><enumeration value="prepared-received"></enumeration></pre>
	<pre><enumeration value="prepared-received"></enumeration> <enumeration value="inaccessible"></enumeration></pre>
	<pre><enumeration value="prepared-received"></enumeration> <enumeration value="inaccessible"></enumeration> <enumeration value="unknown"></enumeration></pre>
	<pre><enumeration value="prepared-received"></enumeration></pre>

<element n<="" th=""><th>ame="inferior-state" substitutionGroup="btp:message"></th></element>	ame="inferior-state" substitutionGroup="btp:message">		
	exType>		
<s< td=""><td>equence></td></s<>	equence>		
	<pre><element <="" name="target-additional-information" pre=""></element></pre>		
ype="btp:addi	tional-information minOccurs="0"/>		
	<pre><element inferior-identifier"="" name="superior-identifier" status"="" type="btp:identi;</pre></td></tr><tr><td></td><td><pre><element name="></element></pre>		
	<simpletype></simpletype>		
	<restriction base="string"></restriction>		
	<pre><enumeration value="active"></enumeration></pre>		
	<pre><enumeration value="inaccessible"></enumeration></pre>		
	<pre><enumeration value="unknown"></enumeration></pre>		
	<pre><element name="reply-requested" type="boolean"></element></pre>		
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>		
</td <td>sequence></td>	sequence>		
	ttribute name="id" type="ID"/>		
<td>lexType></td>	lexType>		
<compl< th=""><th>ame="redirect" substitutionGroup="btp:message"> exType></th></compl<>	ame="redirect" substitutionGroup="btp:message"> exType>		
<compl< td=""><td></td></compl<>			
<compl <s< td=""><td>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/></element></td></s<></compl 	exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/></element>		
<compl <s cype="btp:addi</s </compl 	<pre>exType> equence> <element minoccurs="0" name="target-additional-information" tional-information"=""></element> <element btp:addi<="" name="superior-identifier" td="" type="btp:identi</pre></td></tr><tr><td><compl
<s
cype="><td><pre>exType> equence> <</pre></td></element></pre>	<pre>exType> equence> <</pre>	
<compl <s :ype="btp:addi</s </compl 	<pre>exType> equence> < <element minoccurs="0" name="target-additional-information" tional-information"=""></element> <element btp:addi<br="" name="superior-identifier" type="btp:identi ></pre></td></tr><tr><td><pre><compl.
<s
:ype=">ninOccurs="0"/</element></pre>	exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element btp:addi<br="" inferior-identifier"="" name="superior-identifier" type="btp:identi</td></tr><tr><td><pre><compl.
<s
cype=">ainOccurs="0"/ /></element></element>	<pre>exType> equence> </pre> <pre> equence> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>
<compl. <s ype="btp:addi ninOccurs="0"/ /></s </compl. 	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/></element></element></pre>		
<pre><compl. <s cype="btp:addi ninOccurs="0"/ /> maxOccurs="unbolic complements/ </s </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <="" name="new-address" pre="" type="btp:address"></element></element></element></pre>		
<pre><compl. <s cype="btp:addi ninOccurs="0"/ /> /> maxOccurs="unbolic complements)</s </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/></element></element></element></pre>		
<pre><compl. <s cype="btp:addi ninOccurs="0"/ /> /> maxOccurs="unbolic complements)</s </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <="" name="new-address" pre="" type="btp:address"></element></element></element></pre>		
<pre><compl. <s cype="btp:addi ninOccurs="0"/ /> naxOccurs="unb maxOccurs="unb <!--/</td--><td><pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" neme="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence></element></element></element></element></pre></td></s </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" neme="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence></element></element></element></element></pre>		
<pre> <compl <s cype="btp:addi ninOccurs="0"/ /> naxOccurs="unb naxOccurs="unb <!--<br--><!--<br--><a< td=""><td><pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre></td></a<></s </compl </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre>		
<pre><compl. <s cype="btp:addi ainOccurs="0"/ /> /> haxOccurs="unb maxOccurs="unb <!--<br--><!--<br--><a <td><pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" neme="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence></element></element></element></element></pre></td></a </s </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" neme="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence></element></element></element></element></pre>		
<pre><compl <s ype="btp:addi inOccurs="0"/ /> /> haxOccurs="unb haxOccurs="unb <!--<br--><!--<br--><a< td=""><td><pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre></td></a<></s </compl </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre>		
<pre><compl <s type="btp:addi ninOccurs="0"/ /> naxOccurs="unb naxOccurs="unb <!--<br--><a< td=""><td><pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre></td></a<></s </compl </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre>		
<pre><compl. <s cype="btp:addi ainOccurs="0"/ /> /> haxOccurs="unb maxOccurs="unb <!--<br--><!--<br--><a <td><pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre></td></a </s </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre>		
<pre> <compl. <s cype="btp:addi ainOccurs="0"/ /> /> maxOccurs="unb maxOccurs="unb <!--/<br--><a </a </s </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/></element></element></element></element></pre>		
<pre><compl. <s cype="btp:addi ninOccurs="0"/ /> /> naxOccurs="unb naxOccurs="unb </s </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/> lexType></element></element></element></pre>		
<pre> <compl. <si cype="btp:addi ninOccurs="0"/ /> /> naxOccurs="unb naxOccurs="unb <!--/<br-->aaxOccurs="unb <!--/<br--><a <element n.<br=""><compl< td=""><td><pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/> lexType> ame="begin" substitutionGroup="btp:message"></element></element></element></pre></td></compl<></element></a </si </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/> lexType> ame="begin" substitutionGroup="btp:message"></element></element></element></pre>		
<pre><compl. <s; cype="btp:addi ninOccurs="0"/ /> /> naxOccurs="unb maxOccurs="unb complete <complete </complete </s; </compl. </pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element btp:addi<br="" inferior-identifier"="" name="superior-identifier" type="btp:identi</td></tr><tr><td><pre><compl.
<s
cype=">ninOccurs="0"/ /> naxOccurs="unb maxOccurs="unb <!--/<br-->comp </element> <element n.<br=""><compl <s< pre=""></s<></compl </element></element></pre>	<pre>exType> equence> <element <br="" name="target-additional-information">tional-information" minOccurs="0"/> <element <br="" inferior-identifier"="" name="superior-identifier" old-address"="" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element <br="" name="new-address" type="btp:address">ounded"/> <element minoccurs="0" ref="btp:qualifiers"></element> sequence> ttribute name="id" type="ID"/> lexType> exType> equence></element></element></element></element></pre>	

	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<pre><attribute name="id" type="ID"></attribute></pre>
	<element name="begun" substitutiongroup="btp:message"></element>
	<complextype></complextype>
	<sequence></sequence>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
t	<pre>ype="btp:additional-information" minOccurs="0"/></pre>
	<pre><element <="" name="decider-address" pre="" type="btp:address"></element></pre>
m	inOccurs="0" maxOccurs="unbounded"/>
	<pre><element <="" name="transaction-identifier" pre=""></element></pre>
t	<pre>ype="btp:identifier" minOccurs="0"/></pre>
	<pre><element 0"="" name="inferior-handle" type="btp:identifier'</pre></td></tr><tr><td>m</td><td>inOccurs="></element></pre>
	<pre><element <="" name="inferior-address" pre="" type="btp:address"></element></pre>
m	inOccurs="0" maxOccurs="unbounded"/>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<pre><attribute name="id" type="ID"></attribute></pre>
	<complextype> <sequence></sequence></complextype>
	<pre><complextype> <sequence> <element <="" name="target-additional-information" pre=""></element></sequence></complextype></pre>
t	<pre><complextype> <sequence> <sequence="target-additional-information" minoccurs="0" ype="btp:additional-information"></sequence="target-additional-information"></sequence></complextype></pre>
	<pre><complextype> </complextype></pre> <pre></pre>
	<pre><complextype> </complextype></pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>
m	<pre> <complextype></complextype></pre>
	<pre> <complextype></complextype></pre>
	<pre><complextype> </complextype></pre> <pre> </pre> <pre< td=""></pre<>
	<pre></pre>
	<pre><complextype> </complextype></pre> <pre> </pre> <p< td=""></p<>
	<pre><complextype> </complextype></pre> <pre></pre> <pre></pre> <pre></pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>
	<pre><complextype> </complextype></pre> <pre></pre> <pre> </pre> <pre< td=""></pre<>
	<pre> <complextype> <sequence> <element minoccurs="0" name="target-additional-information" ype="btp:additional-information"></element></sequence></complextype></pre>
	<pre><complextype> </complextype></pre> <pre> </pre>
	<pre> <complextype> <sequence> <element minoccurs="0" name="target-additional-information" ype="btp:additional-information"></element></sequence></complextype></pre>
	<pre><complextype> <sequence> <lelement <br="" name="target-additional-information">ype="btp:additional-information" minOccurs="0"/> <lelement <br="" name="reply-address" type="btp:address">inOccurs="0"/> <lelement <br="" name="transaction-identifier">ype="btp:identifier"/> <lelement minoccurs="0" name="inferiors-list"> <complextype> <lelement minoccurs="0" name="inferiors-list"> <lelement minoccurs="0" name="inferiors-list"> <lelement <br="" name="inferior-handle">ype="btp:identifier" maxOccurs="unbounded"/> </lelement></lelement></lelement></complextype></lelement></lelement></lelement></lelement></sequence> </complextype> </pre>
	<pre><complextype> <sequence> <element <br="" name="target-additional-information">ype="btp:additional-information" minOccurs="0"/> <element <br="" name="reply-address" type="btp:address">inOccurs="0"/> <element <br="" name="transaction-identifier">ype="btp:identifier"/> <element minoccurs="0" name="inferiors-list"> <complextype> <complextype> <complextype> <celement <br="" name="inferior-handle">ype="btp:identifier" maxOccurs="unbounded"/> </celement></complextype></complextype></complextype></element></element></element></element></sequence> </complextype> </pre>
	<pre><complextype> <sequence> <element <br="" name="target-additional-information">ype="btp:additional-information" minOccurs="0"/> <element <br="" name="reply-address" type="btp:address">inOccurs="0"/> <element <br="" name="transaction-identifier">ype="btp:identifier"/> <element minoccurs="0" name="inferiors-list"> <complextype> <complextype> <sequence> <sequence> </sequence> </sequence> </complextype> </complextype></element></element></element></element></sequence></complextype></pre>
	<pre> <complextype> <sequence> <element minoccurs="0" name="target-additional-information" ype="btp:additional-information"></element></sequence></complextype></pre>
	<pre><complextype> <sequence> <element <br="" name="target-additional-information">ype="btp:additional-information" minOccurs="0"/> <element <br="" name="reply-address" type="btp:address">inOccurs="0"/> <element <br="" name="transaction-identifier">ype="btp:identifier"/> <element minoccurs="0" name="inferiors-list"> <complextype> <complextype> <sequence> <sequence> </sequence> </sequence> </complextype> </complextype></element></element></element></element></sequence> </complextype></pre>
	<pre> <complextype> <sequence> <element minoccurs="0" name="target-additional-information" ype="btp:additional-information"></element></sequence></complextype></pre>
	<pre> <complextype> <sequence> <element minoccurs="0" name="target-additional-information" ype="btp:additional-information"></element></sequence></complextype></pre>
	<pre><complextype> <sequence> <element <br="" name="target-additional-information">ype="btp:additional-information" minOccurs="0"/> <element <br="" name="reply-address" type="btp:address">inOccurs="0"/> <element <br="" name="transaction-identifier">ype="btp:identifier"/> <element minoccurs="0" name="inferiors-list"> <complextype> <sequence> <sequence> </sequence> </sequence></complextype> </element>ype="btp:identifier" maxOccurs="unbounded"/> </element></element></element></sequence> </complextype> </pre>
	<pre> <sequence> <element minoccurs="0" name="target-additional-information" ype="btp:additional-information"></element></sequence></pre>

	<pre><element <="" name="reply-address" pre="" type="btp:address"></element></pre>
-	minOccurs="0"/>
Ì	<pre><element <="" name="transaction-identifier" pre=""></element></pre>
ł	type="btp:identifier"/>
	<pre><element minoccurs="0" name="inferiors-list"></element></pre>
	<pre><complextype></complextype></pre>
	<pre><sequence></sequence></pre>
	<pre><element <="" name="inferior-handle" pre=""></element></pre>
	type="btp:identifier" maxOccurs="unbounded"/>
	<pre><element name="report-hazard" type="boolean"></element></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<attribute name="id" type="ID"></attribute>
ļ	
ļ	
ļ	
	<pre><element name="transaction-confirmed" pre="" substitutiongroup="btp:mess</pre></td></tr><tr><td></td><td><complexType></td></tr><tr><td></td><td><sequence></td></tr><tr><td></td><td><pre><element name=" target-additional-information"<=""></element></pre>
	type="btp:additional-information" minOccurs="0"/>
	<pre><element <="" name="transaction-identifier" pre=""></element></pre>
	type="btp:identifier"/>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<pre></pre>
	<attribute name="id" type="ID"></attribute>
	<pre><element dditional_information"<="" name="cancel-transaction" pre="" substitutiongroup="btp:message</pre></td></tr><tr><td></td><td><pre><complexType></pre></td></tr><tr><td></td><td><pre></td></tr><tr><td>ļ</td><td><pre><element name=" target-additional-information"="" turno-"http:=""></element></pre>
1	<pre>type="btp:additional-information" minOccurs="0"/></pre>
	<pre><element minoccurs="0" name="reply-address" type="btp:address"></element></pre>
	<pre><element name="transaction-identifier" type="btp:identifier"></element></pre>
	<pre>cype="btp:identifier"/></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
ļ	<pre></pre>
ļ	<pre></pre>
ļ	
	colomont nemo-"gengel inferienc" substitution Guess "http://www.
J	<pre><element name="cancel-inferiors" substitutiongroup="btp:message"></element></pre>
[<pre><complextype></complextype></pre>
l	<pre><sequence></sequence></pre>
	<pre><element minoccurs="0" name="target-additional-information" type="btp:additional-information"></element></pre>

	<pre><element name="reply-address" s="0" type="btp:address"></element></pre>
	<pre><element <="" name="transaction-identifier" pre=""></element></pre>
type="bt	p:identifier" minOccurs="0"/>
	<pre><element name="inferiors-list"></element></pre>
	<complextype></complextype>
	<sequence></sequence>
	<pre><element <="" name="inferior-handle" pre=""></element></pre>
type="bt	p:identifier" maxOccurs="unbounded"/>
	<element minoccurs="0" ref="btp:qualifiers"></element>
	<attribute name="id" type="ID"></attribute>
<td>ement></td>	ement>
<ele< td=""><td>ment name="transaction-cancelled" substitutionGroup="btp:mes</td></ele<>	ment name="transaction-cancelled" substitutionGroup="btp:mes
	<complextype></complextype>
	<pre><sequence></sequence></pre>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
cype="bt	p:additional-information minOccurs="0"/>
	<pre><element <="" name="transaction-identifier" pre=""></element></pre>
zype="bt	p:identifier"/>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<pre><attribute name="id" type="ID"></attribute></pre>
<td>ement></td>	ement>
	mont nemo-"monuport infersion statuses"
	ment name="request-inferior-statuses"
substitu	tionGroup="btp:message">
	<complextype></complextype>
	<pre></pre>
	Selement name="target-additional-intormation"
type="bt	p:additional-information" minOccurs="0"/>
	<pre>p:additional-information" minOccurs="0"/></pre>
	<pre>p:additional-information" minOccurs="0"/></pre>
	<pre>p:additional-information" minOccurs="0"/></pre>
type="bt ninOccur	<pre>p:additional-information" minOccurs="0"/></pre>
	<pre>p:additional-information" minOccurs="0"/></pre>
	<pre>p:additional-information" minOccurs="0"/></pre>
ninOccur	<pre>p:additional-information" minOccurs="0"/></pre>

	<complextype></complextype>
	<sequence></sequence>
	<pre><element <="" name="target-additional-information" pre=""></element></pre>
:ype=	="btp:additional-information" minOccurs="0"/>
	<pre><element <="" name="responders-identifier" pre=""></element></pre>
ype=	="btp:identifier"/>
	<pre><element name="status-list"></element></pre>
	<complextype></complextype>
	<sequence></sequence>
	<pre><element <="" maxoccurs="unbound" name="status-item" pre=""></element></pre>
	<complextype></complextype>
	<sequence></sequence>
	<pre><element <="" name="inferior-handle" pre=""></element></pre>
ype=	="btp:identifier"/>
	<pre><element name="status"></element></pre>
	<simpletype></simpletype>
	<restriction base="string"></restriction>
	<pre><enumeration value="active"></enumeration></pre>
	<pre><enumeration value="resigned"></enumeration></pre>
	<pre><enumeration value="preparing"></enumeration></pre>
	<pre><enumeration value="prepared"></enumeration></pre>
	<pre><enumeration)<="" pre="" value="autonomously-confirm"></enumeration></pre>
	<pre><enumeration confirming"="" value="autonomously-cancel</pre></td></tr><tr><td></td><td><pre><enumeration value="></enumeration></pre>
	<pre><enumeration value="confirmed"></enumeration></pre>
	<pre><enumeration value="cancelling"></enumeration></pre>
	<pre><enumeration value="cancelled"></enumeration></pre>
	<pre><enumeration)<="" pre="" value="cancel-contradiction"></enumeration></pre>
	<pre><enumeration)<="" pre="" value="confirm-contradiction"></enumeration></pre>
	<pre><enumeration value="hazard"></enumeration></pre>
	<pre><enumeration value="invalid"></enumeration></pre>
	<pre><element minoccurs<="" pre="" ref="btp:qualifiers"></element></pre>
	<pre><element minoccurs="0" ref="btp:qualifiers"></element></pre>
	<pre></pre>
	<pre><attribute name="id" type="ID"></attribute></pre>
<	<pre></pre>
(aab	1ema>

	XML schema for standard qualifiers
<	?xml version="1.0"?>
_	schema
	xmlns="http://www.w3.org/2001/XMLSchema"
	targetNamespace="urn:oasis:names:tc:BTP:qualifiers"
	xmlns:btpq="urn:oasis:names:tc:BTP:qualifiers"
i	xmlns:btp="urn:oasis:names:tc:BTP:xml"
ł	elementFormDefault="qualified">
	Crementrormberdure- quarried >
	<pre><element <="" name="transaction-timelimit" pre=""></element></pre>
2	ubstitutionGroup="btp:qualifier">
	<complextype></complextype>
	<pre><complexcontent></complexcontent></pre>
	<pre><extension base="btp:qualifier-type"></extension></pre>
	<sequence></sequence>
	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
t	ype="nonNegativeInteger"/>
	<pre><element btp:qualifier-type"="" name="inferior-timeout" substitutiongroup="btp:quali</th></tr><tr><td>-</td><td><pre></td></tr><tr><td>-</td><td><pre><extension base="></element></pre>
_	<pre></pre>
+	ype="nonNegativeInteger"/>
	<pre>vpe= nonwegativeInteger /> <element name="intended-decision"></element></pre>
	<pre></pre>
	<pre></pre>
	<pre><restriction base="string"> </restriction></pre>
	<pre></pre>
	<pre></pre>
	<pre></pre> //pimpicipicipici/
	<pre></pre>
	<pre></pre>
J	<pre>//complexContent>//complexType></pre>
J	
J	
	<pre><element <="" name="minimum-inferior-timeout" pre=""></element></pre>
_	<pre>vubstitutionGroup="btp:qualifier"></pre>
	<pre>complexType></pre>
-	
	<pre><complextype> </complextype></pre> <pre><complexcontent> </complexcontent></pre> <pre><extension base="btp:qualifier-type"></extension></pre>

4794	<pre><element <="" name="minimum-timeout" pre=""></element></pre>
4795	type="nonNegativeInteger"/>
4796	
4797	
4798	
4799	
4800	
4801	
4802	<pre><element name="inferior-name" substitutiongroup="btp:qualifier"></element></pre>
4803	<complextype></complextype>
4804	<complexcontent></complexcontent>
4805	<pre><extension base="btp:qualifier-type"></extension></pre>
4806	<sequence></sequence>
4807	<pre><element name="inferior-name" type="string"></element></pre>
4808	<pre></pre>
4809	<pre></pre>
4810	<pre></pre>
4811	<pre></pre>
4812	<pre></pre>
4813	
4814	
4815	

4815	
4816	
4817	Carrier Protocol Bindings
4818	
4819	The notion of bindings is introduced to act as the glue between the BTP messages and an
4820	underlying transport. A binding specification must define various particulars of how the BTP
4821	messages are carried and some aspects of how the related application messages are carried.
4822	This document specifies two bindings: a SOAP binding and a SOAP + Attachments binding.
4823 4824	However, other bindings could be specified by the Oasis BTP technical committee or by a third party. For example, in the future a binding might exist to put a BTP message directly on
4824	top of HTTP without the use of SOAP, or a closed community could define their own
4826	binding. To ensure that such specifications are complete, the Binding Proforma defines the
4827	information that must be included in a binding specification.
4828	
4829	Carrier Protocol Binding Proforma
4830	
4831	A BTP carrier binding specification should provide the following information:
4832	
4833 4834	Binding name: A name for the binding, as used in the "binding name" field of BTP addresses (and available for declaring the capabilities of an implementation). Binding
4834	specified in this document, and future revisions of this document have binding names that are
4836	simple strings of letters, numbers and hyphens (and, in particular, do not contain colons).
4837	Bindings specified elsewhere shall have binding names that are URIs. Bindings specified in
4838	this document use numbers to identify the version of the binding, not the version(s) of the
4839	carrier protocol.
4840	
4841 4842	Binding address format: This section states the format of the "binding address" field of a BTP address for this binding. For many bindings, this will be a URL of some kind; for other
4842	bindings it may be some other form
4844	
4845	BTP message representation: This section will define how BTP messages are represented.
4846	For many bindings, the BTP message syntax will be as specified in the XML schema defined
4847	in this document, and the normal string encoding of that XML will be used.
4848	
4849	Mapping for BTP messages (unrelated) : This section will define how BTP messages that
4850	are not related to application messages are sent in either direction between Superior and
4851 4852	Inferior. (i.e. those messages sent directly between BTP actors). This mapping need not be symmetric (i.e. Superior to Inferior may differ to some degree to Inferior to Superior). The
4852	mapping may define particular rules for particular BTP messages, or messages with particular
4854	parameter values (e.g. the FAULT message with "fault-type" "CommunicationFailure" will
4855	typically not be sent as a BTP message). The mapping states any constraints or requirements
4856	on which BTP may or must be bundled together by compounding.
4857	
4858	Mapping for BTP messages related to application messages: This section will define how
4859	BTP messages that are related to application messages are sent. A binding specification may
4860	defer details of this to a particular application (e.g. a mapping specification could just say

- 4861 "the CONTEXT may be carried as a parameter of an application invocation"). Alternatively,
 4862 the binding may specify a general method that represents the relationship between application
 4863 and BTP messages.
- 4865 Implicit messages: This section specifies which BTP messages, if any, are not sent explicitly
 4866 but are treated as implicit in application messages or other BTP messages. This may depend
 4867 on particular parameter values of the BTP messages or the application messages.
- 4869 Faults: The relationship between the fault and exception reporting mechanisms of the carrier
 4870 protocol and of BTP shall be defined. This may include definition of which carrier protocol
 4871 exceptions are equivalent to a FAULT/communication-failure message.
 4872
- 4873 Relationship to other bindings: Any relationship to other bindings is defined in this section.
 4874 If BTP addresses with different bindings are be considered to match (for purposes of identifying the peer Superior/Inferior and redirection), this should be specified here.
- 4877 Limitations on BTP use: Any limitations on the full range of BTP functionality that are
 4878 imposed by use of this binding should be listed. This would include limitations on which
 4879 messages can be sent, which event sequences are supported and restrictions on parameter
 4880 values. Such limitations may reduce the usefulness of an implementation, but may be
 4881 appropriate in certain environments.
- 4883 Other: Other features of the binding, especially any that will potentially affect interoperation
 4884 should be specified here. This may include restrictions or requirements on the use or support
 4885 of optional carrier parameters or mechanisms.
- 4887 **Bindings for request/response carrier protocols**

4864

4868

4876

4882

4888

- 4889 BTP does not generally follow request/response pattern. In particular, on the outcome 4890 relationship either side may initiate a message – this is an essential part of the presume-abort recovery paradigm although it is not limited to recovery cases. However, there are some BTP 4891 4892 messages, especially in the control relationship, that do have a request/response pattern. 4893 Many (potential) carrier protocols (e.g. HTTP) do have a request/response pattern. The 4894 specification of a binding specification to a request/response carrier protocol needs to state 4895 what rules apply – which messages can be carried by requests, which by responses. The simplest rule is to send all BTP messages on requests, and let the carrier responses travel back 4896 4897 empty. This would be inefficient in use of network resources, and possibly inconvenient 4898 when used for the BTP request/response pairs. 4899
- 4900This section defines a set of rules that allow more efficient use of the carrier, while allowing4901the initiator of a BTP request/response pair to ensure the BTP response is sent back on the4902carrier response. These rules are specified in this section to enable binding specifications to4903reference them, without requiring each binding specification to repeat similar information.
- 4905 A binding to a request/response carrier is not required to use these rules. It may define other
 4906 rules.
 4907

4908	Request/response exploitation rules		
4909 4910	These rules allow implementations to use the request and response of the carrier protocol		
4911	efficiently, and, when a BTP request/response exchange occurs, to either treat the		
4912	request/response exchanges of the carrier protocol and of BTP independently, if both sides		
4913	wish, or allow either side to map them closely.		
4914			
4915	Under these rules, an implementation sending a BTP request (i.e. a message, other than		
4916	CONTEXT, which has "reply-address" as a parameter in the abstract message definition), can		
4917	ensure that it and the reply map to a carrier request/response by supplying no value for the		
4918	"reply-address". An implementation receiving such a request is required to send the BTP		
4919	response on the carrier response.		
4920	~		
4921	Conversely, if an implementation does supply a "reply-address" value on the request, the		
4922	receiver has the option of sending the BTP response back on the carrier response, or sending		
4923	it on a new carrier request.		
4924 4925	Within the outcome relationship, apart from ENDOL/ENDOLLED, there is no "reply		
4923 4926	Within the outcome relationship, apart from ENROL/ENROLLED, there is no "reply- address", and the parties know each other's "address-as-superior" and "address-as-inferior".		
4920	Both sides are permitted to treat the carrier request/response exchanges as just opportunities		
4928	for sending messages to the appropriate destination.		
4929	for scheming messages to the appropriate destination.		
4930	The rules:		
4931			
4932	a) A BTP actor may bundle one or more BTP messages and related groups that		
4933	have the same binding address for their target in a single btp:messages and		
4934	transmit this btp:messages element on a carrier protocol request. There is no		
4935	restriction on which combinations of messages and groups may be so bundled,		
4936	other than that they have the same binding address, and that this binding address		
4937	is usable as the destination of a carrier protocol request.		
4938			
4939	b) A BTP actor that has received a carrier protocol request to which it has not yet		
4940	responded, and which has one or more BTP messages and groups whose binding		
4941	address for the target matches the origin of the carrier request may bundle such		
4942	BTP messages in a single btp:messages element and transmit that on the carrier		
4943	protocol response.		
4944 4945	a) A DTD actor that has reactived on a corriger protocol request one or more DTD		
494 <i>3</i> 4946	c) A BTP actor that has received, on a carrier protocol request, one or more BTP messages or related groups that require a BTP response and for which no reply		
4940 4947	address was supplied, must bundle the responding BTP message and groups in a		
4947 4948	btp:messages element and transmit this element on the carrier protocol response		
4948	to the request that carried the BTP request.		
4950	to the request that carried the D II request.		
4951	d) Where only one message or group is to be sent, it shall be contained within a		
4952	btp:messages element, as a bundle of one element.		
4953			

4954 4955 4956 4957 4958 4959 4960 4961 4962 4963 4963 4964 4965 4966	 e) A BTP actor that receives a carrier protocol request carrying BTP messages that do have a reply address, or which initiate processing that produces BTP messages whose target binding address matches the origin of the request, may freely choose whether to use the carrier protocol response for the replies, or to send back an "empty carrier protocol response", and send the BTP replies in a separately initiated carrier protocol request. The characteristics of an "empty carrier protocol response" shall be stated in the particular binding specification. f) A BTP actor that sends BTP messages on a carrier protocol request must be able to accept returning BTP messages on the corresponding carrier protocol response and, if the actor has offered an address on which it will receive carrier requests, must be able to accept "replying" BTP messages on a separate carrier protocol request.
4967	SOAD Dinding
4968 4969	SOAP Binding
4909 4970 4971 4972 4973 4974	This binding describes how BTP messages will be carried using SOAP as in the <u>SOAP 1.1</u> specification, using the SOAP literal messaging style conventions. If no application message is sent at the same time, the BTP messages are contained within the SOAP Body element. If application messages are sent, the BTP messages are contained in the SOAP Header element.
4975	Binding name: soap-http-1
4976	Diffund func. soap-nup-1
4977 4978	Binding address format: shall be a URL, of type HTTP.
4979 4980 4981 4982	BTP message representation: The string representation of the XML, as specified in the XML schema defined in this document shall be usedThe BTP XML messages are embedded in the SOAP message without the use of any specific encoding rules (literal style SOAP message); hence the encodingStyle attribute need not be set or can be set to an empty string.
4983 4984 4985	Mapping for BTP messages (unrelated): The "request/response exploitation" rules shall be used.
4986 4987 4988 4989 4990	BTP messages sent on an HTTP request or HTTP response which is not carrying an application message, the messages are contained in a single btp:messages element which is the immediate child element of the SOAP Body element.
4991 4992 4993 4994 4995	 An "empty carrier protocol response" sent after receiving an HTTP request containing a btp:messages element in the SOAP Body and the implementation BTP actor chooses just to reply at the lower level (and when the request/response exploitation rules allow an empty carrier protocol response), shall be any of: a) an empty HTTP response
4996 4997 4998 4999	 a) an empty fiff if response b) an HTTP response containing an empty SOAP Envelope c) an HTTP response containing a SOAP Envelope containing a single, empty btp:messages element.

5000 5001 5002 5003 5004 5005	The receiver (the initial sender of the HTTP request) shall treat these in the same way – they have no effect on the BTP sequence (other than indicating that the earlier sending did not cause a communication failure.)
5006 5007 5008 5009 5010	If an application message is being sent at the same time, the mapping for related messages shall be used, as if the BTP messages were related to the application message. (There is no ambiguity in whether the BTP messages are related, because only CONTEXT and ENROL can be related to an application message.)
5010 5011 5012 5013 5014 5015	Mapping for BTP messages related to application messages: All BTP messages sent with an application message, whether related to the application message or not, shall be sent in a single btp:messages element in the SOAP Header. There shall be precisely one btp:messages element in the SOAP Header.
5015 5016 5017 5018	The "request/response exploitation" rules shall apply to the BTP messages carried in the SOAP Header, as if they had been carried in a SOAP Body, unrelated to an application message, sent to the same binding address.
5019 5020 5021	Note – The application protocol itself (which is using the SOAP Body) may use the SOAP RPC or document approach – this is determined by the application.
5022 5023 5024 5025 5026	Only CONTEXT and ENROL messages are related (&) to application messages. If there is only one CONTEXT or one ENROL message present in the SOAP Header, it is assumed to be related to the whole of the application message in the SOAP Body. If there are multiple CONTEXT or ENROL messages, any relation of these BTP messages shall be indicated by application specific means.
5027 5028 5029	Note 1 – An application protocol could use references to the ID values of the BTP messages to indicate relation between BTP CONTEXT or ENROL messages and the application message.
5030 5031	Note 2 However indicated, what the relatedness means, or even whether it has any significance at all, is a matter for the application.
5032 5033 5034 5035 5036 5037 5038 5039 5040	 Implicit messages: A SOAP FAULT, or other communication failure received in response to a SOAP request that had a CONTEXT in the SOAP Header shall be treated as if a CONTEXT_REPLY/repudiated had been received. See also the discussion under "other" about the SOAP mustUnderstand attribute. Faults: A SOAP FAULT or other communication failure shall be treated as FAULT/communication-failure.

5041	Relationship to other bindings: A BTP address for Superior or Inferior that has the binding		
5042	string "soap-http-1" is considered to match one that has the binding string "soap-attachments-		
5043	http-1" if the binding address and additional information fields match.		
5044			
5045	Limitations on BTP use: None		
	Limitations on DTF use. None		
5046			
5047	Other: The SOAP BTP binding does not make use of SOAPAction HTTP header or actor		
5048	attribute. The SOAPAction HTTP header is left to be application specific when there are		
5049	application messages in the SOAP Body, as an already existing web service that is being		
5050	upgraded to use BTP might have already made use of SOAPAction. The SOAPAction HTTP		
5051	header shall be omitted when the SOAP message carries only BTP messages in the SOAP		
5052	Body.		
5053			
5054	The SOAP mustUnderstand attribute, when used on the btp:messages containing a BTP		
5055	CONTEXT, ensures that the receiver (server, as a whole) supports BTP sufficiently to		
5056	determine whether any enrolments are necessary and replies with CONTEXT_REPLY as		
5057	appropriate. The sender of the CONTEXT (and related application message) can use this to		
5058	ensure that the application work is performed as part of the business transaction, assuming the		
5059	receiver's SOAP implementation supports the mustUnderstand attribute. If mustUnderstand if		
5060	false, a receiver can ignore the CONTEXT (if BTP is not supported there), and no		
5061	CONTEXT_REPLY will be returned. It is a local option on the sender (client) side whether		
5062	the absence of a CONTEXT_REPLY is assumed to be equivalent to aCONTEXT_REPLY/ok		
5063	(and the business transaction allowed to proceed to confirmation).		
5065 5064	(and the business transaction and weal to proceed to committation).		
5065	Note – some SOAP implementations may not support the mustUnderstand attribute sufficiently to		
5065	enforce these requirements.		
5060 5067	Example scenario using SOAP binding		
5067	Example scenario using SOAL binding		
	The example below shows on application request with CONTEXT massage cant from		
5069	The example below shows an application request with CONTEXT message sent from		
5070	client.example.com (which includes the Superior) to services.example.com (Service).		
5071 5072			
5072 5073	<soap:envelope< td=""></soap:envelope<>		
5075 5074	<pre>xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"</pre>		
5075	soap:encodingStyle="-">		
5076			
5077	<soap:header></soap:header>		
5078	-		
5079	<pre><btp:messages xmlns:btp="urn:oasis:names:tc:BTP:xml"></btp:messages></pre>		
5080	<pre><btp:context superior-type="atom"></btp:context></pre>		
5081	<pre><btp:superior-address></btp:superior-address></pre>		
5082	<pre><btp:binding>soap-http-1</btp:binding></pre>		
5083	<pre></pre>		
5084 5085	address>http://client.example.com/soaphandleraddress>		
5085 5086	address> <btp:additional-information>btpengine</btp:additional-information>		
5080 5087	information>		
5088	<pre>//btp:superior-address></pre>		
2 3 6 6			

Page 137 of 147

5089	<pre><btp:superior-< pre=""></btp:superior-<></pre>
5090	<pre>identifier>http://example.com/1001</pre>
5091	<pre><btp:qualifiers></btp:qualifiers></pre>
5092	<pre><btpq:transaction-timelimit< pre=""></btpq:transaction-timelimit<></pre>
5093	<pre>xmlns:btpq="urn:oasis:names:tc:BTP:qualifiers"><btpq:timelimit>180</btpq:timelimit></pre>
5094	0
5095	
5096	
5097	
5098	
5099	
5100	
5101	<soap:body></soap:body>
5102	
5103	<nsl:ordergoods< td=""></nsl:ordergoods<>
5104	<pre>xmlns:ns1="http://example.com/2001/Services/xyzgoods"></pre>
5105 5106	<pre><custid>ABC8329045</custid> </pre>
5107	<pre><itemid>224352</itemid> </pre>
5108	<quantity>5</quantity>
5109	
5110	
5111	
5112	
5112	<pre></pre>
5114	
	xample below shows CONTEXT_REPLY and a related ENROL message sent from
	es.example.com to client.example.com, in reply to the previous message. There is no
	ation response, so the BTP messages are in the SOAP Body. The ENROL message
	tot contain the target-additional-information, since the grouping rules for
	TEXT_REPLY & ENROL omit the target address (the receiver of this example
	nbers the superior address from the original CONTEXT)
5121	
5122	<soap:envelope< th=""></soap:envelope<>
5123	<pre>xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"</pre>
5124	<pre>soap:encodingStyle=""></pre>
5125	
5126	<soap:header></soap:header>
5127	
5128 5129	
5130	<soap:body></soap:body>
5131	
5132	<pre><btp:messages xmlns:btp="urn:oasis:names:tc:BTP:xml"> <btp:related-group></btp:related-group></btp:messages></pre>
5133	
5134	_ btp:context-reply>
5135	<pre></pre>
5136	additional-information>
5137	<pre></pre>
5138	<pre></pre>
5139	http://client.example.com/soaphandler
5140	<pre></pre>

Page 138 of 147

5141	<pre></pre>	
5142	btpengine	
5143	<pre></pre>	
5144		
5145	<pre><btp:superior-< pre=""></btp:superior-<></pre>	
5146	<pre>identifier>http://example.com/1001</pre>	
5147	<pre><completion-status>related</completion-status></pre>	•
5148		
5149		
5150	<pre><btp:enrol reply-requested="false"></btp:enrol></pre>	
5151	<pre></pre>	
5152	information>btpengine	
5153	<pre></pre>	
5154	http://example.com/1001	
5155	<pre></pre> //btp:superior-identifier>	
5156	<pre> <</pre>	I
5157	<pre></pre>	
5158	<pre></pre>	
5159	http://services.example.com/soaphandler	
5160		
5160		
5162	<pre> <</pre>	
5162	http://example.com/AAAB	I
5165		I
5165		
5165		
5160	(htp://plated.group)	1
		I
5168		
5168 5160	<pre>/htp:magaagaga</pre>	
5169		
5169 5170		
5169 5170 5171		
5169 5170 5171 5172		
5169 5170 5171 5172 5173		
5169 5170 5171 5172 5173 5174		
5169 5170 5171 5172 5173 5174 5175		
5169 5170 5171 5172 5173 5174		1
5169 5170 5171 5172 5173 5174 5175 5176	 	1
5169 5170 5171 5172 5173 5174 5175 5176 5177		1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178	 SOAP + Attachments Binding	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the <u>SOAP</u>	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the <u>SOAP</u> <u>Messages with Attachments</u> specification. It is a superset of the Basic SOAP binding, soap-	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the <u>SOAP</u>	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the <u>SOAP</u> <u>Messages with Attachments</u> specification. It is a superset of the Basic SOAP binding, soap-	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent.	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the <u>SOAP</u> <u>Messages with Attachments</u> specification. It is a superset of the Basic SOAP binding, soap-	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent. Binding name: soap-attachments-http-1	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184 5185	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent.	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184	SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent. Binding name: soap-attachments-http-1 Binding address format: as for soap-http-1	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184 5185 5186	SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent. Binding name: soap-attachments-http-1 Binding address format: as for soap-http-1	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184 5185 5186 5187	 SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent. Binding name: soap-attachments-http-1	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184 5185 5186 5187 5188	SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent. Binding name: soap-attachments-http-1 Binding address format: as for soap-http-1 BTP message representation: As for soap-http-1	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184 5185 5186 5187 5188 5188 5189	SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent. Binding name: soap-attachments-http-1 Binding address format: as for soap-http-1 BTP message representation: As for soap-http-1 Mapping for BTP messages (unrelated): As for "soap-http-1", except the SOAP Envelope	1
5169 5170 5171 5172 5173 5174 5175 5176 5177 5178 5179 5180 5181 5182 5183 5184 5185 5186 5187 5188	SOAP + Attachments Binding This binding describes how BTP messages will be carried using SOAP as in the SOAP Messages with Attachments specification. It is a superset of the Basic SOAP binding, soap- http-1. The two bindings only differ when application messages are sent. Binding name: soap-attachments-http-1 Binding address format: as for soap-http-1 BTP message representation: As for soap-http-1	1

5191	specified in <u>SOAP Messages with Attachments</u> specification. If an application message is		
5192	being sent at the same time, the mapping for related messages for this binding shall be used,		
5193	as if the BTP messages were related to the application message(s).		
5194			
5195	Mapping for BTP messages related to application messages: MIME packaging shall be		
5196	used. One of the MIME multipart/related parts shall contain a SOAP Envelope, whose SOAP		
5197	Headers element shall contain precisely one btp:messages element, containing any BTP		
5198	messages. Any BTP CONTEXT in the btp:messages is considered to be related to the		
	application message(s) in the SOAP Body, and to also any of the MIME parts referenced		
5199			
5200	from the SOAP Body (using the "href" attribute).		
5201			
5202	Implicit messages: As for soap-http-1.		
5203			
5204	Faults: As for soap-http-1.		
5205			
5206	Relationship to other bindings: A BTP address for Superior or Inferior that has the binding		
5207	string "soap-http-1" is considered to match one that has the binding string "soap-		
5208	attachements-http-1" if the binding address and additional information fields match.		
5209			
5210	Limitations on BTP use: None		
5210			
5211	Other : As for soap-http-1		
5212 5213	Oner. As for soap-indp-1		
	Example using SOAD , Attachments hinding		
5214	Example using SOAP + Attachments binding		
5215 5216	MINE Monsion: 1 0		
5210 5217	MIME-Version: 1.0 Content-Type: Multipart/Related; boundary=MIME_boundary;		
5218	type=text/xml;		
5210	start="someID"		
5220			
5221	MIME_boundary		
5222	Content-Type: text/xml; charset=UTF-8		
5223	Content-ID: someID		
5224			
5225	xml version='1.0' ?		
5226	<soap:envelope< td=""></soap:envelope<>		
5227	<pre>xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"</pre>		
5228	soap-env:encodingStyle="		
5229	http://schemas.xmlsoap.org/soap/encoding/">		
5230 5231			
5232	<soap:header></soap:header>		
5233	<pre><btp:messages xmlns:btp="urn:oasis:names:tc:BTP:xml"></btp:messages></pre>		
5233	<pre></pre>		
5235	<pre></pre>		
5236	<pre></pre>		
5237	<pre> <</br></br></br></br></br></pre>		
5238	http://client.example.com/soaphandler		
5239			

Page 140 of 147

5240			
5241		<pre><btp:superior-< pre=""></btp:superior-<></pre>	
5242		<pre>identifier>http://example.com/1001</pre>	
5243			
5244			
5245			
5246			
5247			
5248		<soap:body></soap:body>	
5249		<pre><ordergoods href="cid:anotherID"></ordergoods></pre>	
5250			
5250			
5252			
5252 5253			
5253 5254			
		MIME_boundary	
5255		Content-Type: text/xml	
5256		Content-ID: anotherID	
5257			
5258		<nsl:ordergoods< td=""></nsl:ordergoods<>	
5259		<pre>xmlns:ns1="http://example.com/2001/Services/xyzgoods"></pre>	
5260		<custid>ABC8329045</custid>	
5261		<itemid>224352</itemid>	
5262		<quantity>5</quantity>	
5263			
5264			
5265			
5266		MIME_boundary	
5267			
5268			
5208			
5269	Confor	mance	
5070			
5270			
5271	A BTP implementation need not implement all aspects of the protocol to be useful. The level		
5272	of conformance of an implementation is defined by which roles it can support using the		
5273	specifi	specified messages and carrier protocol bindings for interoperation with other	
5274		implementations.	
5275	mpier		
5276	·	ially conformant implementation may implement some roles in a non-interoperable	
5277	way, g	iving that implementation's users comparable proprietary functionality.	
5278			
5279	The fo	llowing Roles and Role Groups are used to define conformance:	
5280			
5200	~	ala Grazzaria Dala	
	R	ole Group Role	

Initiator/Terminator

Initiator Terminator

Cohesive Hub

Factory

OASIS BTPDraft Specification 0.9.1.2, 30 January 2002

	Composer (as Decider and Superior) Coordinator (as Decider and Superior) Sub-composer Sub-coordinator
Atomic Hub	Factory Coordinator Sub-coordinator
Cohesive Superior	Composer (as Superior only) Sub-Composer Coordinator (as Superior only) Sub-coordinator
Atomic Superior	Coordinator (as Superior only)) Sub-coordinator
Participant	Inferior Enroller
	or more Role Groups. The following combinations are profiles, although other combinations or
Conformance Profile	Role Groups
Participant Only	Participant
Atomic	Atomic Superior Participant
Cohesive	Full Superior Participant
Atomic Coordination Hub	Initiator/Terminator Atomic Coordination Hub

Participant

Cohesive	Coordination	Hub

Initiator/Terminator Cohesive Coordination Hub Participant

5286 5287

5288 BTP has several features, such as optional parameters, that allow alternative implementation 5289 architectures. Implementations should pay particular attention to avoid assuming their peers 5290 have made the same implementation options as they have (e.g. an implementation that always 5291 sends ENROL with the same inferior address and with the reply address absent (because the 5292 Inferior in all transactions are dealt with by the same addressable entity), must not assume 5293 that the same is true of received ENROLs)

5294

Part 3. Appendices 5295

5296

5297 5298 5299 These terms seem to be all either not used, or effectively defined elsewhere The glossary is the subject of issue 4

A. Glossary 5300

Message	A datum which is produced and then consumed.
Sender	The producer of a message.
Receiver	The consumer of a message.
Transmission	The passage of a message from a sender to a receiver.
Endpoint	A sender or receiver.
Address	An identifier for an endpoint.
<u>Peer</u>	The other party in a two-party relationship, as in Superior to Inferior, or Sender to Receiver
Carrier Protocol	A protocol which defines how transmissions occur.
Carrier Protocol Address	The address of an endpoint for a particular carrier protocol.
(CPA)	
Business Transaction Protocol Address (BTPA)	A compound address consisting of a mandatory carrier protocol address and an optional opaque suffix. PRF - suffix ? I've used "additional
	information"
Actor	An entity which executes procedures, a software agent.
Application	An actor which uses the Business Transaction Protocol.
Application Message	A message produced by an application and consumed by an application.

Application Endpoint	An endpoint of an application message.
Operation	A procedure which is started by a receiver when a message arrives at it.
Application Operation	An operation which is started when an application message arrives.
Contract	Any rule, agreement or promise which constrains an actor's behaviour and is known to any other actor, and upon which any other knowing actor may rely.
Appropriate	In accordance with a pertinent contract.
Inappropriate	In violation of a pertinent contract.
Service	An actor, which on receipt of an application messages, may start an appropriate application operation. For example, a process which advertises an interface allowing defined RPCs to be invoked by a remote client.
Client	An actor which sends application messages to services.
Effect	The changes induced by the incomplete or complete processing of a set of procedures by an actor, which are observable by another contemporary or future actor, and which are made in conformance with a contract known to any such observer. This contract must state the countereffect of the effect, and is known as the countereffect contract. An effect is Completed when the change-inducing processing of the set of procedures is finished. [Need an indirect or consequential damage exclusion clause]
	<i>PRF</i> - Sentence about countereffect contract doesn't fit well
Ineffectual	Describes a set of procedures which has no effect.
Countereffect	An appropriate effect intended to counteract a prior effect.

Countereffect Contract	The contract which governs the relationship between the effect and the countereffect of a procedure. In the absence of any other overriding contracts the countereffect contract is the promise that
	"The Countereffect will attempt so far as is possible to reverse or cancel the Effect such that an observer (on completion of the Countereffect) is unaware that the Effect ever occurred, but this attempt cannot be guaranteed to succeed".
Cancel	Process a countereffect for the current effect of a set of procedures.
Confirm	Ensure that the effect of a set of procedures is completed.
Prepare	Ensure that of a set of procedures is capable of being successfully instructed to cancel or to confirm.
Outcome	A decision to either cancel or confirm.
Participant	A set of procedures which is capable of receiving instructions from a coordinator to prepare, cancel and confirm. A participant must also have a BTPA to which these instructions will be delivered, in the form of BTP messages. A participant is identified by a participant identifier.
Inferior Identifier	An identifier assigned to an Inferior which is unique within the scope of an Address-as-Inferior.
Atomic Business Transaction or Atom	A set of participants (which may have only one member), all of which will receive instructions that will result in a homogeneous outcome. (Transitively, a set of operations, whose effect is capable of countereffect.) An atom is identified by an atom identifier.
Atom Identifier	A globally unique identifier assigned to an atom. <i>PRF – abs msgs define as unambiguous</i> <i>in scope of its address-as-superior, I</i> <i>think.</i>

Coordinator	An actor which decides the outcome of a single atom, and has a lifetime which is coincident with that of the atom. A coordinator can issue instructions to a participant to prepare, cancel and confirm. These instructions take the form of BTP messages. A coordinator is identified by its atom's atom identifier. A coordinator must also have a BTPA to which participants can send BTP messages.
Address-as-Superior	The address used to communicate with an actor playing the role of an Superior
Address-as-Composer	The address used to communicate with a Composer by an application actor that controls its resolution. The messages that might be sent to or received from this endpoint are undefined.
Address-as-Inferior	The address used to communicate with an actor playing the role of an Inferior.
Identity-as-Superior	The combination of Superior Identifier and Address-as-Superior of a given Superior.
Identity-as-Inferior	The combination of Inferior Identifier and Address-as-Inferior of a given Inferior.

5302