

Web Services Atomic Transaction Interop Scenarios (WS-AT Interop Scenarios) 1.1

Working Draft **08**, August **14**, 2006

Document Identifier:

wstx-wsat-1.1-interop_scenarios-wd-08

Location:

http://docs.oasis-open.org/ws-tx/wstx-wsat-1.1-interop_scenarios-wd-08.pdf

Technical Committee:

OASIS WS-TX TC

Chair(s):

Eric Newcomer, Iona
Ian Robinson, IBM

Editor(s):

Ram Jeyaraman, Microsoft

Abstract:

This document defines message flows and formats designed to test interoperability between implementations of the WS-AT specification. Several coordination scenarios have been defined that exercise the critical message exchanges defined in the WS-AT specification.

Status:

This document is a working draft. This document is non-normative. The purpose is to assist with interop testing.

This document was last revised or approved by the WS-TX TC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at www.oasis-open.org/committees/ws-tx.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (www.oasis-open.org/committees/ws-tx/ipr.php).

The non-normative errata page for this specification is located at www.oasis-open.org/committees/ws-tx.

Deleted: 7

Deleted: 02

Deleted: 7

Deleted: 7

Deleted: 7

Deleted: 45

Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS President.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS President.

Copyright © OASIS Open 2006. *All Rights Reserved.*

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself does not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the 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 to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Deleted: 7

Deleted: 45

Table of Contents

- 1 Introduction4
 - 1.1 Scope4
 - 1.2 Test Approach4
 - 1.3 Namespaces4
 - 1.4 Asynchronous Messaging5
 - 1.5 Terminology5
 - 1.6 Normative References5
 - 1.7 Non-Normative References5
- 2 Test Application6
 - 2.1 General Flow6
 - 2.2 Roles used in Test Scenarios7
- 3 Test Scenarios8
 - 1. Test Scenarios for Completion Protocol8
 - Scenario #1.1. CompletionCommit8
 - Scenario #1.2. CompletionRollback8
 - 2. Basic Test Scenarios for 2PC Protocol9
 - Scenario #2.1. Commit9
 - Scenario #2.2. Rollback9
 - 3. Scenarios with two 2PC participants10
 - Scenario #3.1. Phase2Rollback10
 - Scenario #3.2. Readonly11
 - Scenario #3.3. VolatileAndDurable11
 - 4. Early Notifications12
 - Scenario #4.1. EarlyReadonly12
 - Scenario #4.2. EarlyAborted13
 - 5. Test Scenarios with a Simple Failure Model14
 - Scenario #5.1. ReplayCommit14
 - Scenario #5.2. RetryPreparedCommit15
 - Scenario #5.3. RetryPreparedAbort16
 - Scenario #5.4. RetryCommit17
 - Scenario #5.5. PreparedAftertimeout18
 - Scenario #5.6. LostCommitted20
 - 6. Message Snapshots21
- Appendix A. Example of Test Orchestration40
- Appendix B. Test Coverage42
- Appendix C. Acknowledgements43
- Appendix D. Revision History44
- Appendix E. Non-normative Text45

Deleted: 7
Deleted: 45

1 Introduction

1.1 Scope

This document defines message flows and formats designed to test interoperability between implementations of the WS-AT specification. Several coordination scenarios have been defined that exercise the critical message exchanges defined in the WS-AT specification.

Protocol Interoperability testing includes any message exchange that leverages the protocol. Following the workshop model, there are two concurrent mechanisms vendors can use to achieve interoperability: face-to-face (f2f) interop events and online interop testing. Given the time constraints for test execution during a f2f and the number of participants, it is important to prioritize tests to select only those focused on mainline scenarios and which most heavily exercise the protocol. The rest of the protocol testing can be left for online interoperability and in-house QA by each vendor.

We propose the following principles for test selection for online AT interop testing:

1. *Positive tests only.* The failure scenarios used in testing MUST not include invalid states/invalid messages by implementations.
2. *Target Interaction.* Test Scenarios should verify the interaction between two parties according to the protocol.
3. *Target Protocol.* Target scenarios described by WS-AT and WS-Coordination specifications.
4. *Include composition with SSL.*

1.2 Test Approach

The goal of the interop tests in this document is to

Exercise each type of message that is sent or received (excluding faults), as described by WS-AT and WS-Coordination.

Cover the positive paths on the state diagram for both the Completion and 2PC protocols between participants and coordinator.

1.3 Namespaces

The following namespaces are used in this document

Prefix	Namespace	Description
wsa	http://www.w3.org/2005/08/addressing	WS-Addressing
wscoor	http://docs.oasis-open.org/ws-tx/wscoor/2006/06	WS-Coordination
wsat	http://docs.oasis-open.org/ws-tx/wsat/2006/06	WS-AT
tns	http://fabrikam123.com	Test Application
footx	urn:foo-at-extension	sample WS-AT extensions URI

Deleted: W

Field Code Changed

Deleted: W

Field Code Changed

Deleted: W

Field Code Changed

Deleted: T

Deleted: F

Deleted: 7

Deleted: 45

34 1.4 Asynchronous Messaging

35 All messages used in the interop workshop scenarios below correspond to the asynchronous port types
36 described in the WSDL provided in the specifications.

37 1.5 Terminology

38 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
39 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described
40 in **Error! Reference source not found.**

41 1.6 Normative References

42 1.7 Non-Normative References

- 43 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
44 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 45 **[SOAP]** W3C Note, "SOAP: Simple Object Access Protocol 1.1,"
46 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>, 08 May 2000.
- 47 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI):
48 Generic Syntax," RFC 2396, <http://www.ietf.org/rfc/rfc2396.txt>, MIT/LCS, U.C.
49 Irvine, Xerox Corporation, August 1998.
- 50 **[XML-ns]** W3C Recommendation, "Namespaces in XML,"
51 <http://www.w3.org/TR/1999/REC-xml-names-19990114>, 14 January 1999.
- 52 **[XML-Schema1]** W3C Recommendation, "XML Schema Part 1: Structures,"
53 <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>, 2 May 2001.
- 54 **[XML-Schema2]** W3C Recommendation, "XML Schema Part 2: Datatypes,"
55 <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502>, 2 May 2001.
- 56 **[WSADDR]** Web Services Addressing (WS-Addressing) 1.0, W3C Recommendation,
57 <http://www.w3.org/2005/08/addressing>.
- 58 **[WSCOORD]** Web Services Coordination (WS-Coordination)
59 <http://docs.oasis-open.org/ws-tx/wscoor/2006/06>.
- 60 **[WSAT]** Web Services Atomic Transaction (WS-AtomicTransaction)
61 <http://docs.oasis-open.org/ws-tx/wsata/2006/06>.
- 62 **[WSDL]** Web Services Description Language (WSDL) 1.1
63 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.
- 64 **[WSPOLICY]** Web Services Policy Framework (WS-Policy),
65 <http://schemas.xmlsoap.org/ws/2004/09/policy>, VeriSign, Microsoft, Sonic
66 Software, IBM, BEA Systems, SAP, September 2004.
- 67 **[WSSec]** OASIS Standard 200401, March 2004, "Web Services Security: SOAP Message
68 Security 1.0 (WS-Security 2004)", [http://docs.oasis-open.org/wss/2004/01/oasis-
69 200401-wss-soap-message-security-1.0.pdf](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf).
- 70 **[WSSecPolicy]** Web Services Security Policy Language (WS-SecurityPolicy),
71 <http://schemas.xmlsoap.org/ws/2005/07/securitypolicy>, Microsoft, VeriSign, IBM,
72 and RSA Security Inc., July 2005.
- 73 **[WSSecConv]** Web Services Secure Conversation Language (WS-SecureConversation),
74 <http://schemas.xmlsoap.org/ws/2005/02/sc>, OpenNetwork, Layer7, Netegrity,
75 Microsoft, Reactivity, IBM, VeriSign, BEA Systems, Oblix, RSA Security, Ping
76 Identity, Westbridge, Computer Associates, February 2005.
- 77 **[WSTrust]** Web Services Trust Language (WS-Trust),
78 <http://schemas.xmlsoap.org/ws/2005/02/trust>, OpenNetwork, Layer7, Netegrity,
79 Microsoft, Reactivity, VeriSign, IBM, BEA Systems, Oblix, RSA Security, Ping
80 Identity, Westbridge, Computer Associates, February 2005.
- 81

Deleted:

Deleted: .

Deleted:

Field Code Changed

Deleted: 7

Deleted: 45

2 Test Application

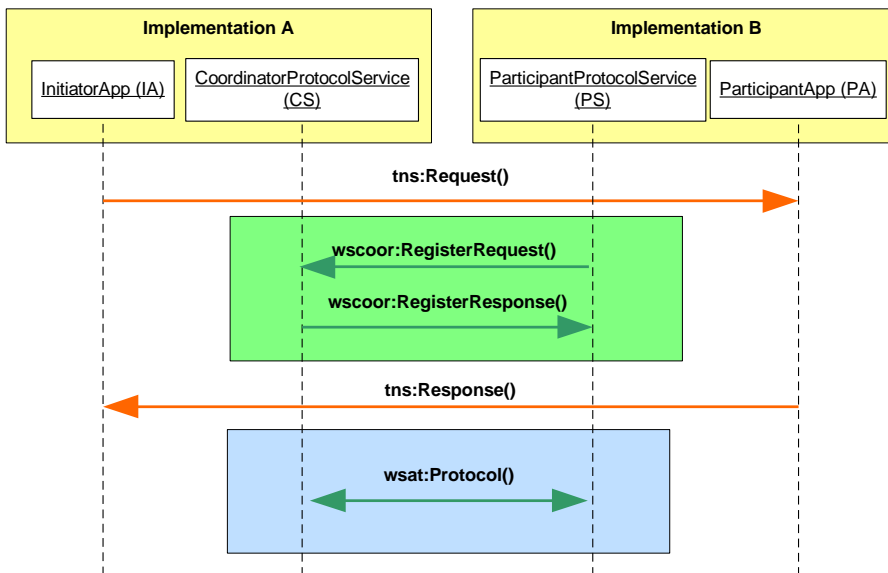
To facilitate interoperability testing, and drive the scenarios, a Test Application is defined. The application comprises a Initiator Application (IA) role and a Participant Application (PA) role, each implemented using different implementations. WSDL for application-level interactions between an IA and PA is defined below.

An interoperability test of two (or more) implementations can be defined by assigning the role of IA to one implementation and the role of PA to all other implementations. Each implementation should support both roles.

2.1 General Flow

This section describes the detailed message flow common for all scenarios in this document. In the test scenarios descriptions such common details will be omitted for brevity.

Implementation A should accept as input the URL of implementation B. Implementation A then initiates a coordinated activity using an internal/private Activation service to create a CoordinationContext (external Activation services are not required and are not tested.) The CoordinationContext includes the EPR of implementation A's Registration Service and a unique coordination identifier (the optional expiration time of the activity should not be specified.) From this point forward, the implementations should interact as follows:



In the first phase, the two implementations agree on the scenario being tested and exchange EPRs for the coordination services being tested.

The IA sends an ApplicationRequest message to the PA. This message includes a [WS-Addressing](#) "Action" header, which specifies the URI of the scenario being tested, and a WS-Coordination "CoordinationContext" header, which identifies the coordinated activity. The action URI for each scenario

109 is derived from the Test Application WSDL (see appendix), using the Default Action Pattern defined in
110 [WS-Addressing](#).

111 The PA joins the coordinated activity by sending a Register message to the Requester's Registration
112 Service. The Register message targets the Registration Service Endpoint Reference specified in the
113 CoordinationContext, and specifies the EPR of PA's Protocol service. The coordination protocol(s) that
114 the PA registers for is scenario specific.

115 The IA's Registration service enlists the PA in the coordination activity, and responds with a
116 RegisterResponse message correlated with the Register message using RelatesTo. The response
117 message includes the EPR of the IA's Coordinator service.

118 The PA sends an Application Response message tns:Response to the IA indicating that the first phase
119 has completed.

120 All Completion and 2PC protocol messages are delivered according to an asynchronous, one-way
121 message exchange pattern. Detail message descriptions are provided below in Section 7.
122

123 In the second phase, the IA and PA execute according to the selected scenario. Messages flows for each
124 scenario are defined below in Sections 2, 3 & 4. In these flows, coordination protocol messages utilize the
125 endpoint references (as defined in [WS-Addressing](#)) exchanged during the registration (Register &
126 RegisterResponse) to direct processing to the coordinator or participant instance. All Completion and
127 2PC protocol messages are delivered according to an asynchronous, one-way message exchange
128 pattern.

129 **2.2 Roles used in Test Scenarios**

130 The test scenarios will use the following roles

- 131 1. Initiator Application (IA)
- 132 2. Participant Application (PA)
- 133 3. Coordinator Service (CS)
- 134 4. Participant Protocol Service (PS, PS1, PS2)

135

136 Each implementation participating in the test should be exercised in each role.

137 3 Test Scenarios

138

139 Note: Even though the WS-Coordination [WSCOOR] specification allows use of anonymous
140 (<http://www.w3.org/2005/08/addressing/anonymous>) wsa:ReplyTo message header element in
141 accordance with WS-Addressing 1.0 Core specification [WSADDR], the test scenario endpoints MUST
142 use addressable non-anonymous wsa:ReplyTo addresses for Register and CreateCoordinationContext
143 messages (part of Registration and Activation service message exchanges).

144 1. Test Scenarios for Completion Protocol

145 The test scenarios in this section use a slightly modified version of the test application. Participant
146 Application PA uses coordination service of IA to create a transaction and then register for Completion
147 protocol.

148

149 Scenarios in this section rely on Initiator's coordinator support of the Activation Service. Testing these
150 scenarios during the interop workshop is optional.

151 Scenario #1.1. CompletionCommit.

152 *Description:* PA creates and commits a Transaction on the IA's Coordinator CS..

153 Initialization

- 154 1. IA sends app message tns:CompletionCommit to PA, that contains endpoint address of the
- 155 Initiator's Application Coordination Service (CS) in the Body.
- 156 2. PA sends app message tns:Response to IA

157 Message Exchange

- 158 3. PA sends wscor:CreateCoordinationContext(wsat) to CS
- 159 4. CS sends wscor:CreateCoordinationContextResponse to PA
- 160 5. PA registers PS for Completion
- 161 6. PA initiates commit using PS
- 162 7. PS sends Completion::Commit to CS
- 163 8. CS sends Completion::Committed to PS.

164 Success Criteria

- 165 CS receives Completion::Commit from PS
- 166 PS receives Completion::Committed from CS

167 Scenario #1.2. CompletionRollback

168 *Description:* PA creates and rolls-back a Transaction on the IA's Coordinator CS.

169 Initialization

- 170 1. IA sends app message tns:CompletionRollback to PA, that contains endpoint address of the
- 171 Initiator's Application Coordination Service (CS) in the Body.
- 172 2. PA sends app message tns:Response to IA

173 Message Exchange

- 174 3. PA sends wscor:CreateCoordinationContext(wsat) to CS

Deleted: 7

Deleted: 45

- 175 4. CS sends wscoor:CreateCoordinationContextResponse to PA
176 5. PA registers PS for Completion
177 6. PS sends Completion::Rollback to CS.
178 7. CS sends Completion::Aborted to PS.

179 **Success Criteria**

- 180 CS receives Completion::Rollback from PS
181 PS receives Completion::Aborted from CS

182 **2. Basic Test Scenarios for 2PC Protocol**

183 Common steps performed by IA locally to create a coordination context creation and enlist for Completion
184 protocol will be omitted throughout this section.

185 **Scenario #2.1. Commit**

186 *Description:* PA registers for Durable2PC, IA initiates Commit, transaction is committed successfully per
187 2PC protocol.

188 **Initialization**

- 189 1. IA sends application message tns:Commit to PA
190 2. PA registers PS with CS for the Durable2PC protocol
191 3. PA sends app message tns:Response to IA.

192 **Message Exchange**

- 193 (IA initiates Commit)
194 4. CS sends Durable2PC::Prepare to PS
195 5. PS sends Durable2PC::Prepared to CS
196 6. CS sends Durable2PC::Commit to PS
197 (CS Committed)
198 7. PS sends Durable2PC::Committed to CS

199 **Success Criteria**

200 CS is in Committed state, receives Durable2PC::Committed from PS.

201 **State Diagram Coverage of the 2PC protocol**

202 Scenario covers the following (State, Event):
203 Coordinator: (Active, Register), (Preparing, Prepared), (Committing, Committed)
204 Participant: (None, RegisterResponse), (Active, Prepare), (PreparedSuccess, Commit)
205

206 **Scenario #2.2. Rollback**

207 *Description:* This scenario tests the common case of transaction rollback initiated by the Initiator
208 Application (IA).

209 **Initialization**

- 210 1. IA sends tns:Rollback to PA
211 2. PA registers PS for Durable2PC on CS
212 3. PA sends tns:Response to IA

Deleted: 7

Deleted: 45

213 **Message Exchange**

- 214 (IA initiates Rollback)
- 215 (CS aborts transaction)
- 216 4. CS sends Durable2PC::Rollback to PS
- 217 5. PS sends Durable2PC::Aborted to CS

218 **Success Criteria**

- 219 CS aborts the transaction
- 220 PS receives Rollback

221 **State Diagram Coverage of the 2PC protocol**

- 222 Scenario covers the following (State, Event):
- 223 Coordinator: (*Active, Register*), (*Aborting | None, Aborted*)
- 224 Participant: (*None, RegisterResponse*), (*Active, Rollback*)

225 **3. Scenarios with two 2PC participants**

226

227 **Scenario #3.1. Phase2Rollback**

228 *Description:* This scenario tests the case of the coordinator (CS) aborting a transaction due to an Aborted
229 vote during the prepare phase. Two participants, PS1 and PS2, are used. The first, PS1, registers for
230 Volatile2PC and votes to commit the transaction. The second, PS2, registers for Durable2PC and votes
231 to abort the transaction. The use of the two enlistments is required to verify the outcome of the
232 transaction on the PA side. The use of a Volatile2PC participant is required to guarantee the ordering of
233 Prepare messages.

234 **Initialization**

- 235 1. IA sends application message tns:Phase2Rollback request to PA
- 236 2. PA registers PS1 for Volatile2PC protocol on CS
- 237 3. PA registers PS2 for Durable2PC protocol on CS
- 238 4. PA sends application message tns:Response to IA

239 **Message Exchange**

- 240 (IA initiates Commit)
- 241 5. CS sends Volatile2PC::Prepare to PS1
- 242 (PS1 votes commit)
- 243 6. PS1 sends Volatile2PC::Prepared to CS
- 244 7. CS sends Durable2PC::Prepare to PS2
- 245 (PS2 votes to abort the transaction)
- 246 8. PS2 sends Durable2PC::Aborted to CS
- 247 (Transaction is aborted)
- 248 9. CS sends Volatile2PC::Rollback to PS1
- 249 10. PS1 sends Volatile2PC::Aborted to CS

250

251 The exact sequence of messages can be controlled by an intermediary that is able to delay messages
252 based on their actions and destination.

Deleted: 7
Deleted: 45

253 **Success Criteria:**

254 IC aborted transaction. P receives Durable2PC::RollBack once.

255

256 **State Diagram Coverage of the 2PC protocol**

257 Scenario covers the following (State, Event):

258 Coordinator: (Active, Register), (Preparing, Prepared), (Preparing, Aborted),
259 (Aborting | None, Aborted)

260 Participant: (None, RegisterResponse), (Active, Prepare), (PreparedSuccess, RollBack).

261

262 **Scenario #3.2. Readonly**

263 *Description:* This scenario tests the case of the coordinator (CS) committing a transaction with a read-only
264 participant. Two participants, PS1 and PS2, are used. The first, PS1, registers for Durable2PC and votes
265 read-only. The second, PS2, registers for Durable2PC and votes to commit the transaction.

266 **Initialization**

- 267 1. IA sends application message tns:Readonly request to PA
268 2. PA registers PS1 for Durable2PC protocol on CS
269 3. PA registers PS2 for Durable2PC protocol on CS
270 4. PA sends application message tns:Response to IA

271 **Message Exchange**

272 (IA initiates Commit)

- 273 5. CS sends Durable2PC::Prepare to PS1
274 6. CS sends Durable2PC::Prepare to PS2
275 (PS1 votes read-only, PS2 votes commit)
276 7. PS1 sends Durable2PC::ReadOnly to CS
277 8. PS2 sends Durable2PC::Prepared to CS
278 (CS decides to commit transaction)
279 9. CS sends Durable2PC::Commit to PS2
280 10. PS2 sends Durable2PC::Committed to CS

281 **Success Criteria:**

282 PS2 receives Durable2PC::Commit from CS

283 CS receives Durable2PC::ReadOnly from PS1

284 CS receives Durable2PC::Committed from PS2

285 **State Diagram Coverage of the 2PC protocol**

286 Scenario covers the following (State, Event):

287 Coordinator: (Active, Register), (Preparing, Prepared), (Preparing, ReadOnly), (Committing, Committed)

288 Participant: (None, RegisterResponse), (Active, Prepare),(PreparedSuccess, Commit).

289 **Scenario #3.3. VolatileAndDurable**

290

291 *Description:* This scenario tests registration during the volatile prepare phase. Two participants, PS1 and
292 PS2, are used. The first, PS1, is registered as a Volatile2PC participant. Upon receiving the

Deleted: 7
Deleted: 45

293 Volatile2PC::Prepare message, the participant application registers a second
294 participant, PS2, for Durable2PC. Once this has completed, participant PS1
295 votes ReadOnly.

296 **Initialization**

- 297 1. IA sends an application message tns:VolatileAndDurable request to PA.
 - 298 2. PA registers PS1 with CS for Volatile2PC protocol
 - 299 3. PA replies to IA with tns:Response app message
- 300

301 **Message Exchange**

- 302 (IA initiates Commit)
- 303 4. CS sends Volatile2PC::Prepare to PS1
 - 304 5. PS1 registers PS2 with CS for a Durable2PC protocol
 - 305 6. PS1 sends Volatile2PC::ReadOnly to CS
 - 306 7. CS sends Durable2PC::Prepare to PS2
 - 307 8. PS2 sends Durable2PC::Prepared to CS.
 - 308 9. CS sends Durable2PC::Commit to PS2
- 309 (CS Committed the transaction)
- 310 10. PS2 sends Durable2PC::Committed to CS

311 **Success Criteria**

312 Transaction is Committed
313 AND
314 PS2 receives Durable2PC::Commit.

315 **State Diagram Coverage of the 2PC protocol**

316 Scenario covers the following (State, Event):
317 Coordinator: (Active, Register), (*Preparing, Register*),(Preparing, ReadOnly), (Preparing, Prepared),
318 (Committing, Committed)
319 Participant: (None, RegisterResponse), (Active, Prepare), (PreparedSuccess, Commit)
320

321 **4. Early Notifications**

322 **Scenario #4.1. EarlyReadOnly**

323
324 *Description:* This scenario tests the case of a participant initiated ReadOnly message occurring prior to
325 the prepare phase. Two participants, PS1 and PS2, are used. The first, PS1, is registered as a
326 Volatile2PC participant. The second, PS2, is registered as a Durable2PC participant. Participant PS1
327 sends an unsolicited Volatile2PC::ReadOnly to coordinator CS, after which the responder sends its
328 Application Response message to the Requester.

329 **Initialization**

- 330 1. IA sends an application message tns:EarlyReadOnly request to PA
 - 331 2. PA registers PS1 with CS for Volatile2PC protocol.
 - 332 3. PA registers PS2 with CS for Durable2PC protocol
- 333

334 **Message Exchange**

- 335 4. PS1 sends Volatile2PC::ReadOnly to CS
336 5. PA sends application response message tns:Response to IA.
337 (I initiates Commit)
338 6. CS sends Durable2PC::Prepare to PS2
339 7. PS2 sends Durable2PC::Prepared to CS
340 8. CS sends Durable2PC::Commit to PS2
341 (CS Commits the transaction)
342 9. PS2 sends Durable2PC::Committed to CS

343 **Success Criteria**

- 344 PS2 receives Durable2PC:Commit
345 AND
346 PS1 does not receive Volatile2PC::Commit
347

348 **State Diagram Coverage of the 2PC protocol**

- 349 Scenario covers the following (State, Event):
350 Coordinator: *(Active, Register), (Active, ReadOnly), (Preparing, Prepared), (Committing, Committed)*
351 Participant: *(None, RegisterResponse), (Active, Prepare), (PreparedSuccess, Commit)*

352 **Scenario #4.2. EarlyAborted**

353 *Description:* This scenario tests the case of a participant initiated Aborted message occurring prior to the
354 prepare phase. Two participants, PS1 and PS2, are used. The first, PS1, is registered as a Volatile2PC
355 participant. The second, PS2, is registered as a Durable2PC participant. After registering the Durable2PC
356 participant, the participant sends an unsolicited Volatile2PC::Aborted for PS1.

357 **Initialization**

- 358 1. IA sends an application message tns:EarlyAborted request to PA
359 2. PA registers PS2 with CS for Durable2PC protocol.
360 3. PA registers PS1 with CS for Volatile2PC protocol.

361 **Message Exchange**

- 362
363 4. PS1 sends async Volatile2PC::Aborted to CS
364 5. PA sends application response message tns:Response to IA.
365 6. CS sends Durable2PC::RollBack to PS2
366 (Transaction is aborted)
367 7. PS2 sends Durable2PC::Aborted to CS
368

369 **Success Criteria**

- 370 CS receives Durable2PC:Aborted from PS2
371 AND
372 PS2 receives Durable2PC:Rollback from CS
373 AND

Deleted: 7

Deleted: 45

374 Transaction is aborted.

375

376 **State Diagram Coverage of the 2PC protocol**

377 Scenario covers the following (State, Event):

378 Coordinator: (Active, Register), (*Active, Aborted*), (Aborting | None, Aborted)

379 Participant: (None, RegisterResponse), (Active, Rollback)

380 **5. Test Scenarios with a Simple Failure Model.**

381

382 The following failures are considered in test scenarios in this section

- 383 1. Participant suffers from recoverable failure.
- 384 2. Communication failure causes message loss

385

386 To resolve ambiguity in the outcome of the tests, additional orchestration is done by either the Initiator
387 Application (IA) or Participant Application (PA), depending on the focus of the test. For each of the
388 scenarios, orchestration is described by a state diagram where, depending on the current state, certain
389 protocol messages are being "dropped", i.e. not delivered to their destination.

390

391 **Scenario #5.1. ReplayCommit.**

392

393 *Description:* PA registers PS for Durable2PC, IA initiates commit, PS suffers from internal recoverable
394 failure after receiving Durable2PC::Commit, sends 2PC::Prepared back to Coordinator upon recovery.
395 Transaction is Committed.

396 **Initialization**

- 397 1. IA sends an application message tns:ReplayCommit request to PA containing
398 CoordinationContext.
- 399 2. PA registers PS with CS for a Durable2PC protocol.
- 400 3. PA responds with an application reply tns:Response.

401 **Message Exchange**

402 (IA initiates Commit)

- 403 1. CS sends Durable2PC::Prepare to PS
- 404 2. PS sends Durable2PC::Prepared to CS
- 405 3. CS sends Durable2PC::Commit to PS
406 (PS suffers from internal failure)
407 (PA prevents any re-sent Commit from reaching PS)
- 408 4. Upon recovery, PS sends 2PC::Prepared to CS
- 409 5. CS re-sends Durable2PC::Commit to PS
- 410 6. PS sends Durable2PC::Committed to CS

411

412 **Success Criteria**

413 CS receives 2PC::Prepared (second one) from PS AND CS receives Durable2PC::Committed from PS.

414

Deleted: 7

Deleted: 45

415 **State Diagram Coverage of the 2PC protocol**

416 Scenario covers the following (State, Event):

417 Coordinator (for PS):

418 (Active, Register), (Preparing, Prepared), (Committing, Prepared), (Committing, Committed)

419 Participant (PS): (None, RegisterResponse), (Active, Prepare), (PreparedSuccess, Commit)

420

421 **Scenario #5.2. RetryPreparedCommit**

422 *Description:* This scenario tests recovery from a communication failure during the prepare phase. Once a
423 durable participant (PS1) receives Durable2PC::Prepare message from Coordinator (CS), it attempts to
424 send Durable2PC::Prepared message but this message never reaches CS (is lost). PS1 retries sending
425 Durable2PC::Prepared and succeeds. Transaction is successfully committed.

426 **Test Orchestration**

427 Initiator (IA) is responsible for test orchestration. The following state table describes the orchestration
428 logic dependency on the incoming and outgoing messages.

429

	Start	Failure	Recovered
Prepare	Relay, transition to Failure	Drop	Drop
Prepared	N/A	Drop, transition to Recovered	Relay
Other 2PC messages	N/A	Drop	Relay

430

431 **Initialization**

432 1. I sends application message tns:RetryPreparedCommit to PA

433 2. PA registers PS1 and PS2 for Durable2PC on CS

434 3. PA sends application message tns:Response to IA

435 .

436 **Message Exchange**

437 (IA initiates Commit)

438 4. CS sends Durable2PC::Prepare to PS1

439 5. CS sends Durable2PC::Prepare to PS2

440 PS1 sends Durable2PC::Prepared to CS..

441 (PS1 fails)

442 6. PS2 sends Durable2PC::Prepared to CS

443 (IA prevents this from reaching CS)

444 (PS1 recovers)

445 7. PS1 sends Durable2PC::Prepared to CS // 2nd Prepared received from PS1

446 8. PS2 sends Durable2PC::Prepared to CS // 1st Prepared received from PS2

447 9. CS sends Durable2PC::Commit to PS1

448 10. PS1 sends Durable2PC::Committed to CS

Deleted: 7

Deleted: 45

- 449 11. CS sends Durable2PC::Commit to PS2
- 450 12. PS2 sends Durable2PC::Committed to CS
- 451
- 452

453 **State Diagram Coverage of the 2PC protocol**

454 Scenario covers the following (State, Event):
 455 Coordinator: (Active, Register), (Preparing, Prepared), (PreparedSuccess, Prepared), (Committing,
 456 Committed)
 457 Participant: (None, RegisterResponse), (Active, Prepare), (PreparedSuccess, Comms Timeout),
 458 (PreparedSuccess, Commit)
 459

460 **Scenario #5.3. RetryPreparedAbort.**

461 *Description:* This scenario tests recovery from a communication failure during the prepare phase. Once a
 462 durable participant (PS1) receives Durable2PC::Prepare message from Coordinator (CS), it attempts to
 463 send Durable2PC::Prepared message but this message never reaches CS (is lost). Eventually, the
 464 transaction times out and aborts. PS1 retries sending Durable2PC::Prepared and succeeds. PS1 is then
 465 notified of the outcome.
 466

467 **Test Orchestration**

468 Initiator (IA) is responsible for test orchestration. The following state table describes the orchestration
 469 logic dependency on the incoming and outgoing messages.
 470

	Start	Failure	Recovered
Prepare	Relay, transition to Failure	Drop	Relay
Prepared	N/A	Drop	Relay
Rollback	N/A	Drop, transition to Recovered	Relay
Other 2PC messages	N/A	Drop	Relay

471
 472 **Initialization**

- 473 1. IA sends an application message tns:RetryPreparedAbort request to PA containing
- 474 CoordinationContext.
- 475 2. PA registers PS1 with CS for Durable2PC protocol.
- 476 3. PA responds with application message tns:Response.
- 477

478 **Message Exchange**

- 479 (IA initiates Commit)
- 480 4. CS sends Durable2PC::Prepare to PS.

Deleted: 7
 Deleted: 45

- 481 5. PS sends Durable2PC::Prepared to CS. IA prevents message from reaching CS
 482 (CS may attempt to retry Durable2PC::Prepare to PS1.
 483 IA blocks those messages)
 484 (CS times out transaction)
 485 6. CS sends Durable2PC::Rollback to PS1. IA prevents message from reaching PS1
 486 (From here on communication is restored and IA relays all the messages, except
 487 Durable2PC::Prepare)
 488 7. PS re-sends Durable2PC::Prepared to CS.
 489 8. CS sends Durable2PC::Rollback to PS.
 490 (CS aborted transaction)
 491 9. PS sends Durable2PC::Aborted to CS
 492

493 **Success Criteria**

494 CS receives Durable2PC::Aborted from PS.
 495

496 **State Diagram Coverage of the 2PC protocol**

497 Scenario covers the following (State, Event):

498 Coordinator: (Active, Register), (Preparing, Expires Timeout), (Aborting | None, Prepared), (Aborting |
 499 None, Aborted)

500 Participant: (None, RegisterResponse), (Active, Prepare), (PreparedSuccess, Comms Timeout),
 501 (PreparedSuccess, Rollback)

502 **Scenario #5.4. RetryCommit.**

503 *Description:* This scenario tests recovery from a communication failure during the Commit phase.

504 Coordinator (CS) receives Durable2PC::Prepared vote from a single participant (PS), decides to commit
 505 the transaction and attempts to send Durable2PC::Commit message to PS. However due to
 506 communication failure, message does not reach PS. CS retries sending Durable2PC::Commit and
 507 succeeds. Transaction is successfully committed..

Deleted: 1
 Deleted: 1
 Deleted: 1

508
 509
 510
 511
 512
 513
 514
 515
 516

512 **Test Orchestration**

513 Participant Application (PA) is responsible for test orchestration. The following state table describes the
 514 orchestration logic dependency on the incoming and outgoing messages.

	Start	Failure	Recovered
Prepared	Relay, transition to Failure	Drop	Relay
Commit	N/A	Drop, transition to Recovered	Relay

Deleted: 7
 Deleted: 45

Other 2PC messages	Relay	Drop	Relay
--------------------	-------	------	-------

520

521 **Initialization**

- 522 1. I sends application message tns:RetryCommit to PA
- 523 2. PA registers PS_v for Durable2PC on CS
- 524 3. PA sends application message tns:Response

Deleted: 1

525 **Message Exchange**

- 526 (IA initiates Commit)
- 527 4. CS sends Durable2PC::Prepare to PS_v.
- 528 5. PS_v sends Durable2PC::Prepared to CS
- 529 (CS decides to commit the transaction)
- 530 6. CS sends Durable2PC::Commit to PS_v, PA prevents message from reaching PS_v, simulating
- 531 communication problem.
- 532 (PS_v may attempt to retry Durable2PC::Prepared to CS.
- 533 PA blocks those messages)
- 534 7. CS resends Durable2PC::Commit to PS_v, PA relays the message.
- 535 (From here on communication is restored and PA relays all the messages)
- 536 8. PS_v sends Durable2PC::Committed to CS

Deleted: 1

Deleted: 1

Deleted: 1

Deleted: 1

Deleted: 1

Deleted: 1

Deleted: 1

538 **Success Criteria**

539 CS resends Durable2PC::Commit to PS AND PS sends Durable2PC::Committed to CS.

540

541 **State Diagram Coverage of the 2PC protocol**

542 Scenario covers the following (State, Event):

543 Coordinator: (Active, Register), (Preparing, Prepared), (Committing, Comms Timeout), (Committing,

544 Committed)

545 Participant: (None, RegisterResponse), (Active, Prepare),(PreparedSuccess, Commit)

546

547 **Scenario #5.5. PreparedAftertimeout.**

548 *Description:* This scenario tests recovery from a communication failure during the prepare phase. With a
 549 prepared volatile participant (PS1) and a prepared durable participant (PS2), the coordinator (CS) times-
 550 out and aborts the transaction. When communication is restored, the participants resend their Prepared
 551 messages to the coordinator. The Coordinator responds with wscoor:UnknownTransaction to the volatile
 552 participant (PS1) and Durable2PC::Rollback to the durable participant (PS2)..

553

554 **Test Orchestration**

555 Initiator Application (IA) is responsible for test orchestration. The following state table describes the
 556 orchestration logic dependency on the incoming and outgoing messages.

557

558

Deleted: 7

Deleted: 45

	Start	Failure1	Recovered
Prepare	Relay	Relay	N/A
Prepared	Relay, transition to Failure1	Drop	Relay
Volatile2PC:Rollback	N/A	Drop	Relay
Durable2PC:Rollback	N/A	Drop, transition to Recovered	Relay
Other Coordination/2PC messages	Relay	Drop	Relay

567 **Initialization**

- 568 1. IA sends an application message tns: PreparedAfterTimeout to PA
569 2. PA registers PS1 for Volatile2PC on CS
570 3. PA registers PS2 for Durable2PC on CS
571 4. PA sends application message tns:Response to IA

572 **Message Exchange**

- 573 (IA initiates Commit)
- 574 5. CS sends Volatile2PC::Prepare to PS1
575 6. PS1 sends Volatile2PC::Prepared to CS
576 7. CS sends Durable 2PC::Prepare to PS2
577 8. PS2 sends Durable::2PC::Prepared to CS. IA prevents the message from reaching CS
578 (CS times-out, and aborts transaction)
579 9. CS sends Volatile2PC::Rollback to PS1. IA prevents the message from reaching PS1
580 10. CS sends Durable2PC::Rollback to PS2. IA prevents the message from reaching PS2
581 (From this point, communication is restored, IA relays all the messages)
- 582 11. PS1 resends Volatile2PC::Prepared to CS
583 12. PS2 resends Durable2PC::Prepared to CS
584 13. CS sends wscor:UnknownTransaction fault to PS1
585 14. CS sends Durable2PC::Rollback to PS2
586 15. PS2 sends Durable2PC::Aborted to CS

587 **Success Criteria**

- 588 PS1 receives wscor:UnknownTransaction fault from CS
589 PS2 receives Durable2PC::Rollback from CS
590 CS receives Durable2PC::Aborted from PS2
591

592 **State Diagram Coverage of the 2PC protocol**

- 593 Scenario covers the following (State, Event):
594 Coordinator: (Active, Register), (Preparing, Expires Timeout), (Preparing, Prepared), (None, Prepared),
595 (None, Aborted)

Deleted: 7

Deleted: 45

596 Participant: *(None, RegisterResponse), (Active, Prepare), (PreparedSuccess, Comms Timeout),*
597 *(PreparedSuccess, Rollback)*

598 **Scenario #5.6. LostCommitted**

599 *Description:* This scenario tests the effect of a lost Committed message.

600 **Test Orchestration**

601 Participant Application (PA) is responsible for test orchestration. The following state table describes the
602 orchestration logic dependency on the incoming and outgoing messages.

603

604

	Start	Failure	Recovered
Commit	Relay, transition to Failure	Relay, transition to Restored	Relay
Committed	N/A	Drop	Relay
Other Coordination/2PC messages	Relay	Drop	Relay

605 **Initialization**

- 606 1. IA sends tns:LostCommitted to PA.
607 2. PA registers PS for a Durable2PC protocol on CS.
608 3. PA responds with application message tns:Response

609 **Message Exchange**

- 610 (IA initiates Commit)
- 611 4. CS sends Durable2PC::Prepare to PS
612 5. PS sends Durable2PC::Prepared to CS
613 6. CS sends Durable2PC::Commit to PS
614 (PS commits and forgets the transaction)
615 7. PS sends Durable2PC::Committed to CS.
616 (Committed message is lost; PA drops the Committed message)
617 (CS Communication timeout occurs)
618 8. CS re-sends Durable2PC::Commit to PS
619 9. PS sends Durable2PC::Committed to CS
620

621 **Success Criteria**

- 622 PS receives Durable2PC::Commit from PS
623 CS receives Durable2PC:Committed from PS
624

625 **State Diagram Coverage of the 2PC protocol**

- 626 Scenario covers the following (State, Event):
627 Coordinator: *(Active, Register), (Preparing, Prepared), (Committing, Comms Timeout), (Committing,*
628 *Committed)*

629 Participant: (None, RegisterResponse), (Active, Prepare), (PreparedSuccess, Commit), (None, Commit)

630

631 6. Message Snapshots

632 WS-Coordination Messages

633 Message CreateCoordinationContext

634 Message Parts

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
wsa:MessageID	Mandatory
wsa:ReplyTo	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
Body	
CreateCoordinationContext	Mandatory
Expires	Optional
CoordinationType	Mandatory

635

```
636 Content-Type: text/xml; charset=utf-8
637 SOAPAction: "http://docs.oasis-open.org/ws-tx/wscoor/2006/06/CreateCoordinationContext"
638 Content-Length: 1239
639 Expect: 100-continue
640 Connection: Keep-Alive
641 Host: localhost
642 <?xml version="1.0" encoding="utf-8"?>
643 <soap:Envelope
644   xmlns:wscor="http://docs.oasis-open.org/ws-tx/wscoor/2006/06"
645   xmlns:wsa="http://www.w3.org/2005/08/addressing"
646   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
647   <soap:Header>
648     <wsa:Action soap:mustUnderstand="1">
649       http://docs.oasis-open.org/ws-tx/wscoor/2006/06/CreateCoordinationContext
650     </wsa:Action>
651     <wsa:MessageID soap:mustUnderstand="1">
652       uuid:300bfad6-8011-4b47-81fe-15777523e2bd
653     </wsa:MessageID>
654     <wsa:ReplyTo soap:mustUnderstand="1">
655       <wsa:Address>
656         http://localhost/WSATWorkShop/I.soap
657       </wsa:Address>
658     </wsa:ReplyTo>
659     <wsa:To soap:mustUnderstand="1">
660       http://localhost/WSATWorkShop/IC.soap
661     </wsa:To>
662   </soap:Header>
```

Deleted: 7

Deleted: 45

```

663 <soap:Body>
664   <wscoor:CreateCoordinationContext>
665     <wscoor:Expires>300000</wscoor:Expires>
666     <wscoor:CoordinationType>
667       http://docs.oasis-open.org/ws-tx/wsat/2006/06
668     </wscoor:CoordinationType>
669   </wscoor:CreateCoordinationContext>
670 </soap:Body>
671 </soap:Envelope>

```

672 **Message CreateCoordinationContextResponse**

673 Message Parts

674

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
wsa:RelatesTo	Mandatory
wsa:MessageID	Optional
Body	
CreateCoordinationContextResponse	Mandatory
CoordinationContext	Mandatory
Identifier	Mandatory
CoordinationType	Mandatory
RegistrationService	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
wsa:ReferenceParameters	Optional

675

```

676 Content-Type: text/xml; charset=utf-8
677 SOAPAction: "http://docs.oasis-open.org/ws-tx/wscoor/2006/06/CreateCoordinationContextResponse"
678 Content-Length: 1239
679 Expect: 100-continue
680 Connection: Keep-Alive
681 Host: localhost
682 <?xml version="1.0" encoding="utf-8"?>
683 <soap:Envelope
684   xmlns:wscoor=" http://docs.oasis-open.org/ws-tx/wscoor/2006/06"
685   xmlns:wsa="http://www.w3.org/2005/08/addressing"
686   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
687   xmlns:foctx="urn:foo-at-extension">
688 <soap:Header>
689   <wsa:Action soap:mustUnderstand="1">
690     http://docs.oasis-open.org/ws-tx/wscoor/2006/06/CreateCoordinationContextResponse
691   </wsa:Action>
692   <wsa:MessageID soap:mustUnderstand="1">
693     uuid:300bfad6-8011-4b47-81fe-15777523e2be
694   </wsa:MessageID>
695   <wsa:RelatesTo>
696     uuid:300bfad6-8011-4b47-81fe-15777523e2bd

```

Deleted: 7

Deleted: 45

```

697     </wsa:RelatesTo>
698     <wsa:To soap:mustUnderstand="1">
699         http://localhost/WSATWorkShop/I.soap
700     </wsa:To>
701 </soap:Header>
702 <soap:Body>
703     <wscoor:CreateCoordinationContextResponse>
704     <wscoor:CoordinationContext>
705         <wscoor:Identifier>uuid:0f05758b-1f0d-4248-a911-90f7bd18ae52</wscoor:Identifier>
706     <wscoor:CoordinationType>
707         http://docs.oasis-open.org/ws-tx/wsata/2006/06
708     </wscoor:CoordinationType>
709     <wscoor:RegistrationService>
710         <wsa:Address>
711             http://localhost/WSATWorkShop/IC.soap
712         </wsa:Address>
713         <wsa:ReferenceParameters>
714             <footx:TransactionID>
715                 uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
716             </footx:TransactionID>
717         </wsa:ReferenceParameters>
718     </wscoor:RegistrationService>
719 </wscoor:CoordinationContext>
720 </wscoor:CreateCoordinationContext>
721 </soap:Body>
722 </soap:Envelope>

```

Field Code Changed

723 **Message Register for Completion Protocol**

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
<i>Registration Service Reference parameters</i>	Mandatory if specified in the EPR
<i>Registration Service Reference parameters</i>	Mandatory if specified in the EPR
wsa:MessageID	Mandatory
wsa:ReplyTo	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
Body	
Register	Mandatory
ProtocolIdentifier	Mandatory
ParticipantProtocolService	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
wsa:ReferenceParameters	Optional

Deleted: 7

Deleted: 45

724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777

```
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://docs.oasis-open.org/ws-tx/wscoor/2006/06/Register"
Content-Length: 1239
Expect: 100-continue
Connection: Keep-Alive
Host: localhost
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:wscoor=" http://docs.oasis-open.org/ws-tx/wscoor/2006/06"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:footx="urn:foo-at-extension">
  <soap:Header>
    <wsa:Action soap:mustUnderstand="1">
      http://docs.oasis-open.org/ws-tx/wscoor/2006/06/Register
    </wsa:Action>
    <wsa:MessageID soap:mustUnderstand="1">
      uuid:300bfad6-8011-4b47-81fe-15777523e2bf
    </wsa:MessageID>
    <wsa:ReplyTo soap:mustUnderstand="1">
      <wsa:Address>
        http://localhost/WSATWorkShop/I.soap
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soap:mustUnderstand="1">
      http://localhost/WSATWorkShop/IC.soap
    </wsa:To>
    <footx:TransactionID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
      uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
    </footx:TransactionID>
  </soap:Header>
  <soap:Body>
    <wscoor:Register>
      <wscoor:ProtocolIdentifier>
        http://docs.oasis-open.org/ws-tx/wscoor/2006/06/Completion
      </wscoor:ProtocolIdentifier>
      <wscoor:ParticipantProtocolService>
        <wsa:Address>
          http://localhost/WSATWorkShop/I.soap
        </wsa:Address>
      <wsa:ReferenceParameters>
        <footx:TransactionID>
          uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
        </footx:TransactionID>
        <footx:EnlistmentID>
          2f00e898-0702-4583-b8bd-20876b699e3e
        </footx:EnlistmentID>
      </wsa:ReferenceParameters>
    </wscoor:ParticipantProtocolService>
  </wscoor:Register>
</soap:Body>
</soap:Envelope>
```

Deleted: 6
http://docs.oasis-open.org/ws-tx/wscoor/2006/06/Completion

778 **Message RegisterResponse**

779 Message Parts

780

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory

Deleted: 7

Deleted: 45

wsa:RelatesTo	Mandatory
wsa:MessageID	Optional
Body	
RegisterResponse	Mandatory
CoordinatorProtocolService	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
wsa:ReferenceParameters	Optional

784

```

785 Content-Type: text/xml; charset=utf-8
786 SOAPAction: "http://docs.oasis-open.org/ws-tx/wscoor/2006/06/RegisterResponse"
787 Content-Length: 1239
788 Expect: 100-continue
789 Connection: Keep-Alive
790 Host: localhost
791 <?xml version="1.0" encoding="utf-8"?>
792 <soap:Envelope
793   xmlns:wscor="http://docs.oasis-open.org/ws-tx/wscoor/2006/06"
794   xmlns:wsa="http://www.w3.org/2005/08/addressing"
795   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
796   xmlns:footx="urn:foo-at-extension">
797   <soap:Header>
798     <wsa:Action soap:mustUnderstand="1">
799       http://docs.oasis-open.org/ws-tx/wscoor/2006/06/RegisterResponse
800     </wsa:Action>
801     <wsa:MessageID soap:mustUnderstand="1">
802       uuid:300bfad6-8011-4b47-81fe-15777523e2ba
803     </wsa:MessageID>
804     <wsa:RelatesTo>
805       uuid:300bfad6-8011-4b47-81fe-15777523e2bf
806     </wsa:RelatesTo>
807   </soap:Header>
808   <soap:Body>
809     <wscor:RegisterResponse>
810       <wscor:CoordinatorProtocolService>
811         <wsa:Address>
812           http://localhost/WSATWorkShop/I/soap
813         </wsa:Address>
814         <wsa:ReferenceParameters>
815           <footx:TransactionID>
816             uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
817           </footx:TransactionID>
818           <footx:EnlistmentID>
819             12cab914-ec4d-400c-ac2f-47a9b9228b5b
820           </footx:EnlistmentID>
821         </wsa:ReferenceParameters>
822       </wscor:CoordinatorProtocolService>
823     </wscor:RegisterResponse>
824   </soap:Body>
825 </soap:Envelope>

```

826 **Message Register for Volatile2PC protocol**

827 Same as Register for Completion protocol, except for the protocol identifier.

828 Action used by this message:

829 <http://docs.oasis-open.org/ws-tx/wscoor/2006/06/Register>

830 The Body element of this message:

831 <soap:Body>

Deleted: 7
Deleted: 45

```

832 <wscoor:Register>
833 <wscoor:Protocol>
834   http://docs.oasis-open.org/ws-tx/wsata/2006/06/Volatile2PC
835 </wscoor:Protocol>
836 <wscoor:ParticipantProtocolService>
837   <wsa:Address>
838     http://localhost/WSATWorkShop/I.soap
839   </wsa:Address>
840   <wsa:ReferenceParameters>
841     <footx:TransactionID>
842       uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
843     </footx:TransactionID>
844     <footx:EnlistmentID>
845       2f00e898-0702-4583-b8bd-20876b699e3e
846     </footx:EnlistmentID>
847   </wsa:ReferenceParameters>
848 </wscoor:ParticipantProtocolService>
849 </wscoor:Register>
850 </soap:Body>

```

Deleted: <http://docs.oasis-open.org/ws-tx/wsata/2006/06/Volatile2PC>

851 **Message Register for Durable2PC protocol**

852 Same as Register for Completion protocol, except for the protocol identifier.

853 Action used by this message:

854 <http://docs.oasis-open.org/ws-tx/wscoor/2006/06/Register>

855 The Body element of this message:

```

856 <soap:Body>
857 <wscoor:Register>
858 <wscoor:Protocol>
859   http://docs.oasis-open.org/ws-tx/wsata/2006/06/Durable2PC
860 </wscoor:Protocol>
861 <wscoor:ParticipantProtocolService>
862   <wsa:Address>
863     http://localhost/WSATWorkShop/I.soap
864   </wsa:Address>
865   <wsa:ReferenceParameters>
866     <footx:TransactionID>
867       uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
868     </footx:TransactionID>
869     <footx:EnlistmentID>
870       2f00e898-0702-4583-b8bd-20876b699e3e
871     </footx:EnlistmentID>
872   </wsa:ReferenceParameters>
873 </wscoor:ParticipantProtocolService>
874 </wscoor:Register>
875 </soap:Body>

```

Deleted: <http://docs.oasis-open.org/ws-tx/wsata/2006/06/Durable2PC>

876 **WS-AtomicTransaction Messages, Completion Protocol**

877 **Message Completion::Commit**

878 Message Parts.

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
<i>Coordinator Protocol Service Reference parameters</i>	Mandatory if specified in the CoordinatorProtocolService EPR

Deleted: 7

Deleted: 45

<i>Coordinator Protocol Service Reference parameters</i>	Mandatory if specified in the CoordinatorProtocolService EPR
wsa:MessageID	Optional
wsa:From (Value MUST NOT be http://www.w3.org/2005/08/addressing/anonymous or http://www.w3.org/2005/08/addressing/none)	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
wsa:ReferenceParameters	Optional
wsa:ReplyTo	Mandatory
wsa:Address (Value MUST be http://www.w3.org/2005/08/addressing/none)	Mandatory
Body	
Commit	Mandatory

883

884

```

885 Content-Type: text/xml; charset=utf-8
886 SOAPAction: "http://docs.oasis-open.org/ws-tx/wsac/2006/06/Commit"
887 Content-Length: 1239
888 Expect: 100-continue
889 Connection: Keep-Alive
890 Host: localhost
891 <?xml version="1.0" encoding="utf-8"?>
892 <soap:Envelope
893   xmlns:wscor=" http://docs.oasis-open.org/ws-tx/wscor/2006/06"
894   xmlns:wsa="http://www.w3.org/2005/08/addressing"
895   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
896   xmlns:footx="urn:foo-at-extension">
897 <soap:Header>
898 <wsa:Action soap:mustUnderstand="1">
899   http://docs.oasis-open.org/ws-tx/wsac/2006/06/Commit
900 </wsa:Action>
901 <wsa:From>
902   <wsa:Address>
903     http://localhost/WSATWorkShop/I.soap
904   </wsa:Address>
905   <wsa:ReferenceParameters>
906     <footx:TransactionID wsa:IsReferenceParameter='true'>
907       uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
908     </footx:TransactionID>
909     <footx:EnlistmentID wsa:IsReferenceParameter='true'>
910       2f00e898-0702-4583-b8bd-20876b699e3e
911     </footx:EnlistmentID>
912   </wsa:ReferenceParameters>
913 </wsa:From>
914 <wsa:ReplyTo>
915   <wsa:Address>
916     http://www.w3.org/2005/08/addressing/none
917   </wsa:Address>
918 </wsa:ReplyTo>
919 <wsa:To soap:mustUnderstand="1">
920   http://localhost/WSATWorkShop/IC.soap
921 </wsa:To>
922 <footx:TransactionID wsa:IsReferenceParameter='true'>

```

Deleted: 7
Deleted: 45

```

923         uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
924     </footx:TransactionID>
925     <footx:EnlistmentID wsa:IsReferenceParameter='true'>
926         12cab914-ec4d-400c-ac2f-47a9b9228b5b
927     </footx:EnlistmentID>
928 </soap:Header>
929 <soap:Body>
930     <wsat:Commit xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsat/2006/06"/>
931 </soap:Body>
932 </soap:Envelope>

```

933

934 **Message Completion::Committed**

935 Message Parts

936

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
Participant Protocol Service Reference parameters	Mandatory if specified in the EPR
Participant Protocol Service Reference parameters	Mandatory if specified in the EPR
wsa:MessageID	Optional
wsa:ReplyTo	Mandatory
wsa:Address (Value MUST be http://www.w3.org/2005/08/addressing/none)	Mandatory
Body	
Committed	Mandatory

937

```

938 Content-Type: text/xml; charset=utf-8
939 SOAPAction: "http://docs.oasis-open.org/ws-tx/wsat/2006/06/Committed"
940 Content-Length: 1239
941 Expect: 100-continue
942 Connection: Keep-Alive
943 Host: localhost
944 <?xml version="1.0" encoding="utf-8"?>
945 <soap:Envelope
946     xmlns:wscoor="http://docs.oasis-open.org/ws-tx/wscoor/2006/06"
947     xmlns:wsa="http://www.w3.org/2005/08/addressing"
948     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
949     xmlns:footx="urn:foo-at-extension">
950 <soap:Header>
951     <wsa:Action soap:mustUnderstand="1">
952         http://docs.oasis-open.org/ws-tx/wsat/2006/06/Committed
953     </wsa:Action>
954     <wsa:ReplyTo>
955         <wsa:Address>
956             http://www.w3.org/2005/08/addressing/none
957         </wsa:Address>
958     </wsa:ReplyTo>
959     <wsa:To soap:mustUnderstand="1">
960         http://localhost/WSATWorkShop/I.soap

```

Deleted: 7

Deleted: 45

```

961     </wsa:To>
962     <footx:TransactionID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
963         uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
964     </footx:TransactionID>
965     <footx:EnlistmentID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
966         2f00e898-0702-4583-b8bd-20876b699e3e
967     </footx:EnlistmentID>
968 </soap:Header>
969 <soap:Body>
970     <wsat:Committed xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsac/2006/06"/>
971 </soap:Body>
972 </soap:Envelope>

```

973

974 **Message Completion::Rollback**

975 Message Parts.

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
<i>Coordinator Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
<i>Coordinator Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
wsa:MessageID	Optional
wsa:From (Value MUST NOT be 'http://www.w3.org/2005/08/addressing/anonymous' or 'http://www.w3.org/2005/08/addressing/none')	Mandatory
Wsa:Address	Mandatory
Wsa:ReferenceParameters	Optional
wsa:ReplyTo (Value MUST be 'http://www.w3.org/2005/08/addressing/none')	Mandatory
Wsa:Address	Mandatory
Body	
Rollback	Mandatory

976

977

```

978 Content-Type: text/xml; charset=utf-8
979 SOAPAction: "http://docs.oasis-open.org/ws-tx/wsac/2006/06/Rollback"
980 Content-Length: 1239
981 Expect: 100-continue
982 Connection: Keep-Alive
983 Host: localhost
984 <?xml version="1.0" encoding="utf-8"?>
985 <soap:Envelope
986     xmlns:wscor=" http://docs.oasis-open.org/ws-tx/wsccor/2006/06"
987     xmlns:wsa="http://www.w3.org/2005/08/addressing"

```

Deleted: 7

Deleted: 45

```

988   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
989   xmlns:footx="urn:foo-at-extension">
990 <soap:Header>
991   <wsa:Action soap:mustUnderstand="1">
992     http://docs.oasis-open.org/ws-tx/wsat/2006/06/Rollback
993   </wsa:Action>
994   <wsa:From>
995     <wsa:Address>
996       http://localhost/WSATWorkShop/I.soap
997     </wsa:Address>
998     <wsa:ReferenceParameters>
999       <footx:TransactionID wsa:IsReferenceParameter='true'>
1000         uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
1001       </footx:TransactionID>
1002       <footx:EnlistmentID wsa:IsReferenceParameter='true'>
1003         2f00e898-0702-4583-b8bd-20876b699e3e
1004       </footx:EnlistmentID>
1005     </wsa:ReferenceParameters>
1006   </wsa:From>
1007   <wsa:ReplyTo>
1008     <wsa:Address>
1009       http://www.w3.org/2005/08/addressing/none
1010     </wsa:Address>
1011   </wsa:ReplyTo>
1012   <wsa:To soap:mustUnderstand="1">
1013     http://localhost/WSATWorkShop/IC.soap
1014   </wsa:To>
1015   <footx:TransactionID wsa:IsReferenceParameter='true'>
1016     uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
1017   </footx:TransactionID>
1018   <footx:EnlistmentID wsa:IsReferenceParameter='true'>
1019     12cab914-ec4d-400c-ac2f-47a9b9228b5b
1020   </footx:EnlistmentID>
1021 </soap:Header>
1022 <soap:Body>
1023   <wsat:Rollback xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsat/2006/06"/>
1024 </soap:Body>
1025 </soap:Envelope>

```

1026

1027 **Message Completion::Aborted**

1028 Message Parts

1029

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
<i>Participant Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
<i>Participant Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
wsa:MessageID	Optional
wsa:ReplyTo	Mandatory
wsa:Address (Value MUST be http://www.w3.org/2005/08/addressing/none)	Mandatory

Deleted: 7
 Deleted: 45

Body	
Aborted	Mandatory

1030

```

1031 Content-Type: text/xml; charset=utf-8
1032 SOAPAction: "http://docs.oasis-open.org/ws-tx/wsat/2006/06/Aborted"
1033 Content-Length: 1239
1034 Expect: 100-continue
1035 Connection: Keep-Alive
1036 Host: localhost
1037 <?xml version="1.0" encoding="utf-8"?>
1038 <soap:Envelope
1039   xmlns:wscor="http://docs.oasis-open.org/ws-tx/wscor/2006/06"
1040   xmlns:wsa="http://www.w3.org/2005/08/addressing"
1041   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
1042   xmlns:footx="urn:foo-at-extension">
1043   <soap:Header>
1044     <wsa:Action soap:mustUnderstand="1">
1045       http://docs.oasis-open.org/ws-tx/wsat/2006/06/Aborted
1046     </wsa:Action>
1047     <wsa:ReplyTo>
1048       <wsa:Address>
1049         http://www.w3.org/2005/08/addressing/none
1050       </wsa:Address>
1051     </wsa:ReplyTo>
1052     <wsa:To soap:mustUnderstand="1">
1053       http://localhost/WSATWorkShop/I.soap
1054     </wsa:To>
1055     <footx:TransactionID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
1056       uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
1057     </footx:TransactionID>
1058     <footx:EnlistmentID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
1059       2f00e898-0702-4583-b8bd-20876b699e3e
1060     </footx:EnlistmentID>
1061   </soap:Header>
1062   <soap:Body>
1063     <wsat:Aborted xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsat/2006/06"/>
1064   </soap:Body>
1065 </soap:Envelope>

```

1066 **WS-AtomicTransaction Messages, 2PC Protocols**

1067 Messages for Volatile and Durable 2PC protocols are identical, this paragraph contains Durable2PC
1068 messages.

1069 **Message Durable2PC::Prepare**

1070 Message Parts.

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
<i>Participant Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
<i>Participant Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
wsa:MessageID	Optional

Deleted: 7
 Deleted: 45

wsa:From (Value MUST NOT be 'http://www.w3.org/2005/08/addressing/anonymous' or 'http://www.w3.org/2005/08/addressing/none')	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
wsa:ReferenceParameters	Optional
wsa:ReplyTo (Value MUST be 'http://www.w3.org/2005/08/addressing/none')	Mandatory
wsa:Address	Mandatory
Body	
Prepare	Mandatory

1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116

```
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://docs.oasis-open.org/ws-tx/wssoap/2006/06/Prepare"
Content-Length: 1239
Expect: 100-continue
Connection: Keep-Alive
Host: localhost
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:wscor=" http://docs.oasis-open.org/ws-tx/wscor/2006/06"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:footx="urn:foo-at-extension">
  <soap:Header>
    <wsa:Action soap:mustUnderstand="1">
      http://docs.oasis-open.org/ws-tx/wssoap/2006/06/Prepare
    </wsa:Action>
    <wsa:From>
      <wsa:Address>
        http://localhost/WSATWorkShop/I.soap
      </wsa:Address>
      <wsa:ReferenceParameters>
        <footx:TransactionID wsa:IsReferenceParameter='true'>
          uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
        </footx:TransactionID>
        <footx:EnlistmentID wsa:IsReferenceParameter='true'>
          2f00e898-0702-4583-b8bd-20876b699e3e
        </footx:EnlistmentID>
      </wsa:ReferenceParameters>
    </wsa:From>
    <wsa:ReplyTo>
      <wsa:Address>
        http://www.w3.org/2005/08/addressing/none
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soap:mustUnderstand="1">
      http://localhost/WSATWorkShop/I.soap
    </wsa:To>
    <footx:TransactionID wsa:IsReferenceParameter='true'>
      uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
    </footx:TransactionID>
    <footx:EnlistmentID wsa:IsReferenceParameter='true'>
      12cab914-ec4d-400c-ac2f-47a9b9228b5b
    </footx:EnlistmentID>
  </soap:Header>
```

Deleted: 7
Deleted: 45


```

1117 <soap:Body>
1118   <wsat:Prepare xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsat/2006/06"/>
1119 </soap:Body>
1120 </soap:Envelope>

```

1121 **Message Durable2PC::Prepared**

1122 Message Parts

1123

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
wsa:From (Value MUST NOT be 'http://www.w3.org/2005/08/addressing/anonymous' or 'http://www.w3.org/2005/08/addressing/none')	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
wsa:ReferenceParameters	Optional
wsa:ReplyTo (Value MUST be 'http://www.w3.org/2005/08/addressing/none')	Mandatory
wsa:Address	Mandatory
<i>Coordinator Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
<i>Coordinator Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
wsa:MessageID	Optional
Body	
Prepared	Mandatory

```

1124
1125 Content-Type: text/xml; charset=utf-8
1126 SOAPAction: "http://docs.oasis-open.org/ws-tx/wsat/2006/06/Prepared"
1127 Content-Length: 1239
1128 Expect: 100-continue
1129 Connection: Keep-Alive
1130 Host: localhost
1131 <?xml version="1.0" encoding="utf-8"?>
1132 <soap:Envelope
1133   xmlns:wscor="http://docs.oasis-open.org/ws-tx/wscor/2006/06"
1134   xmlns:wsa="http://www.w3.org/2005/08/addressing"
1135   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
1136   xmlns:foctx="urn:foo-at-extension">
1137 <soap:Header>
1138   <wsa:Action soap:mustUnderstand="1">
1139     http://docs.oasis-open.org/ws-tx/wsat/2006/06/Prepared
1140   </wsa:Action>
1141   <wsa:To soap:mustUnderstand="1">
1142     http://localhost/WSATWorkShop/I.soap

```

Deleted: 7

Deleted: 45

```

1143 </wsa:To>
1144 <wsa:From>
1145   <wsa:Address>
1146     http://localhost/WSATWorkShop/I.soap
1147   </wsa:Address>
1148   <wsa:ReferenceParameters>
1149     <footx:TransactionID wsa:IsReferenceParameter='true'>
1150       uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
1151     </footx:TransactionID>
1152     <footx:EnlistmentID wsa:IsReferenceParameter='true'>
1153       2f00e898-0702-4583-b8bd-20876b699e3e
1154     </footx:EnlistmentID>
1155   </wsa:ReferenceParameters>
1156 </wsa:From>
1157 <wsa:ReplyTo>
1158   <wsa:Address>
1159     http://www.w3.org/2005/08/addressing/none
1160   </wsa:Address>
1161 </wsa:ReplyTo>
1162 <footx:TransactionID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
1163   uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
1164 </footx:TransactionID>
1165 <footx:EnlistmentID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
1166   2f00e898-0702-4583-b8bd-20876b699e3e
1167 </footx:EnlistmentID>
1168 </soap:Header>
1169 <soap:Body>
1170   <wsat:Prepared xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsac/2006/06"/>
1171 </soap:Body>
1172 </soap:Envelope>

```

1173

1174 **Message Durable2PC::Commit**

1175 Message structure is identical to Completion::Commit, except that wsa:To and wsa:From point to
 1176 Participant and Coordinator's protocol service respectively.

1177 **Message Durable2PC::Committed**

1178 Message structure is identical to Completion::Committed, except that wsa:To points to Coordinator's
 1179 protocol service.

1180 **Message Durable2PC::Rollback**

1181 Message structure is identical to Completion::Rollback, except that wsa:To and wsa:From point to
 1182 Participant and Coordinator's protocol service respectively.

1183 **Message Durable2PC::Aborted**

1184 Message structure is identical to Completion::Committed, except that
 1185 wsa:To points to Coordinator's protocol service respectively,
 1186 Action is http://docs.oasis-open.org/ws-tx/wsac/2006/06/Aborted
 1187 Body contains wsat:Aborted element.

1188

1189 **Message Durable2PC::ReadOnly**

1190 Message Parts

1191

Name	Mandatory?
------	------------

Deleted: 7
 Deleted: 45

Header	
wsa:Action	Mandatory
wsa:To	Mandatory
<i>Coordinator Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
<i>Coordinator Protocol Service Reference parameters</i>	Mandatory if specified in the EPR
wsa:MessageID	Optional
wsa:ReplyTo (Value MUST be 'http://www.w3.org/2005/08/addressing/none')	Mandatory
wsa:Address	Mandatory
ReadOnly	Mandatory

1192

```

1193 Content-Type: text/xml; charset=utf-8
1194 SOAPAction: "http://docs.oasis-open.org/ws-tx/wsat/2006/06/ReadOnly"
1195 Content-Length: 1239
1196 Expect: 100-continue
1197 Connection: Keep-Alive
1198 Host: localhost
1199 <?xml version="1.0" encoding="utf-8"?>
1200 <soap:Envelope
1201   xmlns:wscor="http://docs.oasis-open.org/ws-tx/wscor/2006/06"
1202   xmlns:wsa="http://www.w3.org/2005/08/addressing"
1203   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
1204   xmlns:footx="urn:foo-at-extension">
1205   <soap:Header>
1206     <wsa:Action soap:mustUnderstand="1">
1207       http://docs.oasis-open.org/ws-tx/wsat/2006/06/ReadOnly
1208     </wsa:Action>
1209     <wsa:ReplyTo>
1210       <wsa:Address>
1211         http://www.w3.org/2005/08/addressing/none
1212       </wsa:Address>
1213     </wsa:ReplyTo>
1214     <wsa:To soap:mustUnderstand="1">
1215       http://localhost/WSATWorkShop/I.soap
1216     </wsa:To>
1217     <footx:TransactionID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
1218       uuid:ac4c2599-0ad9-4b88-ac33-12b765390929
1219     </footx:TransactionID>
1220     <footx:EnlistmentID soap:mustUnderstand="1" wsa:IsReferenceParameter='true'>
1221       2f00e898-0702-4583-b8bd-20876b699e3e
1222     </footx:EnlistmentID>
1223   </soap:Header>
1224   <soap:Body>
1225     <wsat:ReadOnly xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsat/2006/06"/>
1226   </soap:Body>
1227 </soap:Envelope>

```

1228

1229 **Test Application**

1230 **Request Message to Responder P**

1231 Message Parts

Deleted: 7
Deleted: 45

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
wsa:MessageID	Mandatory
wsa:ReplyTo	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
CoordinationContext	Mandatory
Identifier	Mandatory
CoordinationType	Mandatory
RegistrationService	Mandatory
wsa:Address	Mandatory
wsa:ReferenceParameters	Optional
wsa:ReferenceParameters	Optional
Body	
tns:Commit	Mandatory

1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265

```
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://fabrikam123.com/ParticipantPortType/Commit"
Content-Length: 1239
Expect: 100-continue
Connection: Keep-Alive
Host: localhost
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:wsat="http://docs.oasis-open.org/ws-tx/wsata/2006/06"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsa:Action soap:mustUnderstand="1">
      http://fabrikam123.com/ParticipantPortType/Commit
    </wsa:Action>
    <wsa:MessageID soap:mustUnderstand="1">
      uuid:300bfad6-8011-4b47-81fe-15777523e2cd
    </wsa:MessageID>
    <wsa:ReplyTo soap:mustUnderstand="1">
      <wsa:Address>
        http://localhost/WSATWorkShop/I.soap
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:To soap:mustUnderstand="1">
      http://localhost/WSATWorkShop/P.soap
    </wsa:To>
    <wscoor:CoordinationContext soap:mustUnderstand="1">
      <wscoor:Identifier>uuid:0f05758b-1f0d-4248-a911-90f7bd18ae52</wscoor:Identifier>
    </wscoor:CoordinationContext>
    <wscoor:CoordinationType>
      http://docs.oasis-open.org/ws-tx/wsata/2006/06
    </wscoor:CoordinationType>
    <wscoor:RegistrationService>
      <wsa:Address>http://localhost/WSATWorkShop/IC.soap</wsa:Address>
    </wscoor:RegistrationService>
  </soap:Header>
  <tns:Commit/>
</soap:Envelope>
```

Deleted: <http://docs.oasis-open.org/ws-tx/wsata/2006/06>
Deleted: 7
Deleted: 45

```

1268     </wscoor:RegistrationService>
1269 </wscoor:CoordinationContext>
1270 </soap:Header>
1271 <soap:Body>
1272     <Commit xmlns="http://fabrikam123.com"/>
1273 </soap:Body>
1274 </soap:Envelope>

```

1275 **Response Message to Requester I**

Name	Mandatory?
Header	
wsa:Action	Mandatory
wsa:To	Mandatory
wsa:RelatesTo	Mandatory
wsa:MessageID	Optional
Reference Parameters from I Application EPR	Mandatory if specified in the EPR
Reference Parameters from I Application EPR	Mandatory if specified in the EPR
Body	
tns:Response	Mandatory

```

1276
1277 Content-Type: text/xml; charset=utf-8
1278 SOAPAction: "http://fabrikam123.com/InitiatorPortType/Response"
1279 Content-Length: 1239
1280 Expect: 100-continue
1281 Connection: Keep-Alive
1282 Host: localhost
1283 <?xml version="1.0" encoding="utf-8"?>
1284 <soap:Envelope
1285     xmlns:wsat=" http://docs.oasis-open.org/ws-tx/wsat/2006/06"
1286     xmlns:wsa="http://www.w3.org/2005/08/addressing"
1287     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
1288 <soap:Header>
1289     <wsa:Action soap:mustUnderstand="1">
1290         http://fabrikam123.com/InitiatorPortType/Response
1291     </wsa:Action>
1292     <wsa:MessageID soap:mustUnderstand="1">
1293         uuid:300bfad6-8011-4b47-81fe-15777523e2cd
1294     </wsa:MessageID>
1295     <wsa:RelatesTo>
1296         uuid:300bfad6-8011-4b47-81fe-15777523e2bf
1297     </wsa:RelatesTo>
1298     <wsa:ReplyTo soap:mustUnderstand="1">
1299         <wsa:Address>
1300             http://localhost/WSATWorkShop/I.soap
1301         </wsa:Address>
1302     </wsa:ReplyTo>
1303     <wsa:To soap:mustUnderstand="1">
1304         http://localhost/WSATWorkShop/P.soap
1305     </wsa:To>
1306 </soap:Header>
1307 <soap:Body>
1308     <Response xmlns="http://fabrikam123.com"/>
1309 </soap:Body>
1310 </soap:Envelope>

```

Deleted: 7

Deleted: 45

1311 **Test Application WSDL**

```
1312 <?xml version="1.0" encoding="utf-8"?>
1313 <definitions
1314   xmlns:s="http://www.w3.org/2001/XMLSchema"
1315   xmlns:tns="http://fabrikam123.com"
1316   targetNamespace="http://fabrikam123.com"
1317   xmlns="http://schemas.xmlsoap.org/wsdl/">
1318   <types>
1319     <s:schema elementFormDefault="qualified" targetNamespace="http://fabrikam123.com">
1320       <s:complexType name="TestMessageType">
1321         <s:sequence />
1322       </s:complexType>
1323       <s:element name="Response" type="tns:TestMessageType" />
1324       <s:element name="CompletionCommit" type="tns:TestMessageType" />
1325       <s:element name="CompletionRollback" type="tns:TestMessageType" />
1326       <s:element name="Commit" type="tns:TestMessageType" />
1327       <s:element name="Rollback" type="tns:TestMessageType" />
1328       <s:element name="Phase2Rollback" type="tns:TestMessageType" />
1329       <s:element name="ReadOnly" type="tns:TestMessageType" />
1330       <s:element name="VolatileAndDurable" type="tns:TestMessageType" />
1331       <s:element name="EarlyReadOnly" type="tns:TestMessageType" />
1332       <s:element name="EarlyAborted" type="tns:TestMessageType" />
1333       <s:element name="ReplayCommit" type="tns:TestMessageType" />
1334       <s:element name="RetryPreparedCommit" type="tns:TestMessageType" />
1335       <s:element name="RetryPreparedAbort" type="tns:TestMessageType" />
1336       <s:element name="RetryCommit" type="tns:TestMessageType" />
1337       <s:element name="PreparedAfterTimeout" type="tns:TestMessageType" />
1338       <s:element name="LostCommitted" type="tns:TestMessageType" />
1339     </s:schema>
1340   </types>
1341   <message name="CompletionCommit">
1342     <part name="parameters" element="tns:CompletionCommit" />
1343   </message>
1344   <message name="CompletionRollback">
1345     <part name="parameters" element="tns:CompletionRollback" />
1346   </message>
1347   <message name="Commit">
1348     <part name="parameters" element="tns:Commit" />
1349   </message>
1350   <message name="Rollback">
1351     <part name="parameters" element="tns:Rollback" />
1352   </message>
1353   <message name="Phase2Rollback">
1354     <part name="parameters" element="tns:Phase2Rollback" />
1355   </message>
1356   <message name="ReadOnly">
1357     <part name="parameters" element="tns:ReadOnly" />
1358   </message>
1359   <message name="VolatileAndDurable">
1360     <part name="parameters" element="tns:VolatileAndDurable" />
1361   </message>
1362   <message name="EarlyReadOnly">
1363     <part name="parameters" element="tns:EarlyReadOnly" />
1364   </message>
1365   <message name="EarlyAborted">
1366     <part name="parameters" element="tns:EarlyAborted" />
1367   </message>
1368   <message name="ReplayCommit">
1369     <part name="parameters" element="tns:ReplayCommit" />
1370   </message>
1371   <message name="RetryPreparedCommit">
1372     <part name="parameters" element="tns:RetryPreparedCommit" />
1373   </message>
1374   <message name="RetryPreparedAbort">
1375     <part name="parameters" element="tns:RetryPreparedAbort" />
1376   </message>
1377   <message name="RetryCommit">
1378     <part name="parameters" element="tns:RetryCommit" />
1379   </message>
1380   <message name="PreparedAfterTimeout">
1381     <part name="parameters" element="tns:PreparedAfterTimeout" />
```

Deleted: 7
Deleted: 45

```
1382 </message>
1383 <message name="LostCommitted">
1384   <part name="parameters" element="tns:LostCommitted" />
1385 </message>
1386 <message name="Response">
1387   <part name="parameters" element="tns:Response" />
1388 </message>
1389 <portType name="InitiatorPortType">
1390   <operation name="Response">
1391     <input name="Response" message="tns:Response" />
1392   </operation>
1393 </portType>
1394 <portType name="ParticipantPortType">
1395   <operation name="CompletionCommit"><!-- Scenario 1.1 -->
1396     <input name="CompletionCommit" message="tns:CompletionCommit" />
1397   </operation>
1398   <operation name="CompletionRollback"><!-- Scenario 1.2 -->
1399     <input name="CompletionRollback" message="tns:CompletionRollback" />
1400   </operation>
1401   <operation name="Commit"><!-- Scenario 2.1 -->
1402     <input name="Commit" message="tns:Commit" />
1403   </operation>
1404   <operation name="Rollback"><!-- Scenario 2.2 -->
1405     <input name="Rollback" message="tns:Rollback" />
1406   </operation>
1407   <operation name="Phase2Rollback"><!-- Scenario 3.1 -->
1408     <input name="Phase2Rollback" message="tns:Phase2Rollback" />
1409   </operation>
1410   <operation name="ReadOnly"><!-- Scenario 3.2 -->
1411     <input name="ReadOnly" message="tns:ReadOnly" />
1412   </operation>
1413   <operation name="VolatileAndDurable"><!-- Scenario 3.3 -->
1414     <input name="VolatileAndDurable" message="tns:VolatileAndDurable" />
1415   </operation>
1416   <operation name="EarlyReadOnly"><!-- Scenario 4.1 -->
1417     <input name="EarlyReadOnly" message="tns:EarlyReadOnly" />
1418   </operation>
1419   <operation name="EarlyAborted"><!-- Scenario 4.2 -->
1420     <input name="EarlyAborted" message="tns:EarlyAborted" />
1421   </operation>
1422   <operation name="ReplayCommit"><!-- Scenario 5.1 -->
1423     <input name="ReplayCommit" message="tns:ReplayCommit" />
1424   </operation>
1425   <operation name="RetryPreparedCommit"><!-- Scenario 5.2 -->
1426     <input name="RetryPreparedCommit" message="tns:RetryPreparedCommit" />
1427   </operation>
1428   <operation name="RetryPreparedAbort"><!-- Scenario 5.3 -->
1429     <input name="RetryPreparedAbort" message="tns:RetryPreparedAbort" />
1430   </operation>
1431   <operation name="RetryCommit"><!-- Scenario 5.4 -->
1432     <input name="RetryCommit" message="tns:RetryCommit" />
1433   </operation>
1434   <operation name="PreparedAfterTimeout"><!-- Scenario 5.5 -->
1435     <input name="PreparedAfterTimeout" message="tns:PreparedAfterTimeout" />
1436   </operation>
1437   <operation name="LostCommitted"><!-- Scenario 5.6 -->
1438     <input name="LostCommitted" message="tns:LostCommitted" />
1439   </operation>
1440 </portType>
1441 </definitions>
```

Deleted: 7

Deleted: 45

1442

Appendix A. Example of Test Orchestration

1443 This section describes an example of the solution IA or PA could use for the test orchestration required in
1444 scenarios in Section 5.

1445

1446 Consider Scenario 5.3 as an example. IA is required to perform additional test orchestration for this
1447 scenario. The approach described below introduces an intermediary "Man In the Middle" (MIM) controlled
1448 by IA, to facilitate additional test orchestration of the message exchange between actual Initiator and
1449 Participant(s) protocol and registration services. IA configures such intermediary to send messages to
1450 PA's application, protocol and registration services, and receive messages on behalf of IA.

1451

1452 Man In the Middle (MIM) follows a simple state diagram for each instance of the message exchange
1453 (conversation instance) corresponding to a particular scenario. Depending on what state MIM is in for a
1454 conversation instance it will

1455 change EPRs and relay messages to ensure that subsequent protocol messages for a
1456 particular Scenario communication are sent through the Man In the Middle and carry unique
1457 ID that identifies scenario and particular conversation instance.

1458 Relay messages

1459 Drop messages

1460

1461 For each scenario conversation instance MIM can reside in one of the following three states:

1462 Active – waiting on application message to start conversation instance that corresponds to a
1463 scenario

1464 Init – modifying application and registration messages to ensure that subsequent protocol
1465 messages for a particular Scenario communication are sent through the Man In the Middle
1466 and carry unique ID that identifies scenario and this conversation instance.

1467 Scenario Specific States – relaying or dropping particular protocol messages depending on
1468 scenario orchestration needs

1469

1470 The following State table for a sample Man In The Middle implementation illustrates this behavior:

1471

	Active	Init	Scenario Specific States
Scenario App Message	Change Registration Service EPR to point to MiM and add Ref property to identify scenario and conversation instance. Forward modified message. Transition to Init State.	N/A	N/A
Register	N/A	Change Participant Protocol Service and ReplyTo EPRs to point to MiM and add a reference property to both to identify scenario and conversation instance. Forward modified message	N/A

Deleted: 7

Deleted: 45

RegisterResponse	N/A	Change Coordinator's Protocol Service EPR to point to MiM and add a reference property to both to identify scenario and conversation instance. Forward modified message. Transition to Scenario Specific States.	N/A
Protocol Messages	Relay	Relay	Scenario Specific actions (drop or relay)

1472

1473

1474 If scenario involves SOAP level security and MiM is a standalone component that communicates with
 1475 both parties over wire, updating EPRs may be problematic. In this case MiM needs to have access to all
 1476 keys used in the communication.

Deleted: 7

Deleted: 45

1477

Appendix B. Test Coverage

1478

2PC Protocol State Diagram Coverage

1479

Atomic Transaction 2PC protocol (Coordinator View)							
Inbound Events	States						
	None	Active	Preparing	Prepared	PreparedSuccess	Committing	Aborting
Register	Out of Scope	Covered	Covered	Out of Scope	Out of Scope	Out of Scope	Out of Scope
Prepared	Covered	Out of Scope	Covered	Out of Scope	Out of Scope	Out of Scope	Covered
ReadOnly	Out of Scope	Covered	Covered	Out of Scope	Out of Scope	Out of Scope	Out of Scope
Aborted	Covered	Covered	Covered	Out of Scope	Out of Scope	Out of Scope	Covered
Committed	Out of Scope	Out of Scope	Out of Scope	Out of Scope	Out of Scope	Covered	Out of Scope
Replay	Covered	Out of Scope	Covered	Out of Scope	Out of Scope	Covered	Covered

1480

1481

Atomic Transaction 2PC protocol (Participant View)							
Inbound Events	States						
	None	Active	Preparing	Prepared	PreparedSuccess	Committing	Aborting
Register Response	Covered	Out of Scope	Out of Scope	Out of Scope	Out of Scope	Out of Scope	Out of Scope
Prepare	Out of Scope	Covered	Out of Scope	Out of Scope	Out of Scope	Out of Scope	Out of Scope
Commit	Covered	Out of Scope	Out of Scope	Out of Scope	Covered	Out of Scope	Out of Scope
Rollback	Out of Scope	Covered	Out of Scope	Out of Scope	Covered	Out of Scope	Out of Scope

1482

1483

Deleted: 7

Deleted: 45

1484

Appendix C. Acknowledgements

1485

1486

This document is based on initial contribution to OASIS WS-TX Technical Committee by the

1487

following authors: Thomas Freund, IBM, Kirill Gavrylyuk, Microsoft, Max Feingold, Microsoft, Dave

1488

Langworthy, Microsoft, Thomas Mikalsen, IBM, Stefan Tai, IBM, Ian Robinson, IBM, Tony Storey, IBM.

1489

1490

The following individuals were members of the committee during the development of this

1491

specification:

1492

TBD

1493

1494

[Participant Name, Affiliation | Individual Member]

1495

[Participant Name, Affiliation | Individual Member]

1496

Deleted: 7

Deleted: 45

1497

Appendix D. Revision History

1498

[optional; should not be included in OASIS Standards]

1499

Revision	Date	Editor	Changes Made
01	March 30, 2006	Ram Jeyaraman	Contributed Scenario document has been updated to use the OASIS template.
02	May 31, 2006	Ram Jeyaraman	Updated to reflect resolutions to issues 28 and 30.
03	June 5, 2006	Ram Jeyaraman	Updated to reflect resolutions to issues 39, 52.
04	June 29, 2006	Ram Jeyaraman	Scenarios 2.1, 3.1, 5.1, 5.2 and 5.6 updated. Some textual edits to sections 1.1, 1.2.
05	July 08, 2006	Ram Jeyaraman	Section 3 has been updated to include the requirement to use addressable non-anonymous ReplyTo addresses for Register and CreateCoordinationContext messages. Scenario 5.3 updated. Section 1.7 updated to include reference to WS-Coordination specification.
06	July 20, 2006	Ram Jeyaraman	Editorial change: Scenario 5.1 description reflect the use of participant (PS), instead of two participants (PS1 and PS2).
07	August 02, 2006	Ram Jeyaraman	Replaced namespaces http://docs.oasis-open.org/wstx/wscoor/2006/03 and http://docs.oasis-open.org/wstx/wsac/2006/03 with http://docs.oasis-open.org/wstx/wscoor/2006/06 and http://docs.oasis-open.org/wstx/wsac/2006/06 respectively.
<u>08</u>	<u>August 14, 2006</u>	<u>Ram Jeyaraman</u>	<u>Updated the namespace URIs to point to the correct link.</u>

1500

Deleted: 7

Deleted: 45

1501
1502

Appendix E. Non-normative Text

Deleted: 7
Deleted: 45