

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU

Q.787

(03/93)

SPECIFICATIONS OF SIGNALLING SYSTEM No. 7

TRANSACTION CAPABILITIES (TC) TEST SPECIFICATION

ITU-T Recommendation Q.787 Superseded by a more recent version

(Previously "CCITT Recommendation")

FOREWORD

The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the International Telecommunication Union. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, established the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

ITU-T Recommendation Q.787 was prepared by the ITU-T Study Group XI (1988-1993) and was approved by the WTSC (Helsinki, March 1-12, 1993).

NOTES

As a consequence of a reform process within the International Telecommunication Union (ITU), the CCITT ceased to exist as of 28 February 1993. In its place, the ITU Telecommunication Standardization Sector (ITU-T) was created as of 1 March 1993. Similarly, in this reform process, the CCIR and the IFRB have been replaced by the Radiocommunication Sector.

In order not to delay publication of this Recommendation, no change has been made in the text to references containing the acronyms "CCITT, CCIR or IFRB" or their associated entities such as Plenary Assembly, Secretariat, etc. Future editions of this Recommendation will contain the proper terminology related to the new ITU structure.

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

© ITU 1994

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

			Page		
1	Intro	luction	1		
2	Objec	ctives of the test specification	1		
3	Scope	2	1		
4	Gene	ral principles of test	2		
5	Test environment				
	5.1	Signalling Relation			
	5.2	Configuration	2		
6	Background traffic				
7	Test l	ist	3		
	7.1	TC Transaction sublayer test specification	3		
	7.2	TC Component Sublayer test specification	115		

Recommendation Q.787

TRANSACTION CAPABILITIES (TC) TEST SPECIFICATION

(Helsinki, 1993)

1 Introduction

This Recommendation contains a detailed set of tests for the SS No. 7 Transaction Capabilities (TC). These tests are intended to validate the protocol specified in Recommendations Q.771 to Q.774. This Recommendation conforms to *Blue Book* (1988) which describes the basic rules for a test specification, as specified in Recommendation Q.780.

2 Objectives of the test specification

The objective of the test specification is to provide:

Validation – A level of confidence that a given implementation conforms to the *Blue Book* (1988) Recommendations Q.771 to Q.774 for SS No. 7 TC.

Compatibility – A level of confidence that two implementations of SS No. 7 TC are able to interwork.

The following criteria have been used in the generation of this test specification:

- 1) the test specification does not provide exhaustive testing of all aspects of the SS No. 7 TC;
- 2) all tests are of a practical nature and implementable using the available technology;
- 3) the test list concentrates on the testing of normal signalling procedures. Testing of abnormal signalling procedures are only identified where this is regarded as particularly useful;
- 4) the test list does not include any tests which are application specific. These tests should be contained in application specific testing documentation and are outside the scope of this test specification.

3 Scope

The test scripts are divided into two sections, 7.1 transaction sublayer (TSL) and 7.2 the component sublayer (CSL) tests. Most TSL and CSL functions are dependent on each other and will need to be performed together. The division between TSL and CSL is for clarification and understanding only and does not imply an implementation.

This test specification is designed to verify the TCAP functionality by testing TCAP messages and their contents. Performance aspects such as the limits of numbers of transactions ID's are not taken into account in this test specification.

Some tests in this Recommendation require the generation of primitives, therefore when performing these tests, appropriate normal system actions of the TCAP user will have to be chosen which result in the indicated primitive being generated.

The testing of primitives is outside the scope of this Recommendation. Both messages and primitives are shown in the expected message sequence diagrams as indicated below, but primitives are shown for ease of understanding only.



1

The test description provides a guide for the correct interpretation and implementation of the test, but it does not constrain its realisation. In particular, any reference to the internal structure of the Implementation Under Test (IUT), such as confirmation of internal states of the TC state machines, is given for clarification only and its practical realisation can be application dependent or vary from one test to another. All questions and checks in the test description should be answered "YES" for correct operation.

Throughout the test specification, mention is made of "state machines". This specification conceptual model is used in Recommendation Q.774 to aid understanding. It does not imply an implementation, even when the test script asks for the state to be confirmed at the end of some tests.

Possible methods of ensuring that the software has returned to the required state are enumerated in the "Guidance" 7.1.1 and 7.2.1.

The test specification is independent of any specific application, or implementation.

4 General principles of test

The tests are described as "Validation" or "Validation and Compatibility" tests. Each test script indicates in the "Type of Test" field, whether the test is "VAT" (Validation) or "VAT and CPT" (Validation and Compatibility).

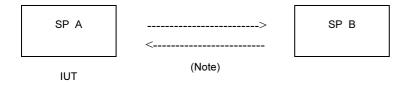
5 Test environment

5.1 Signalling Relation

A stable signalling relation is required between "SP A" and "SP B" in order to test TCAP effectively. A tested network service layer, e.g. MTP and SCCP signalling relation, should be used for compatibility tests.

5.2 Configuration

Only one configuration is required to perform the tests given in the proposed test list, as shown in Figure 1:



NOTE – The arrows indicate a singalling relation.

FIGURE 1/Q.787

Configuration: 1

6 Background traffic

These tests do not take into account any level of background traffic.

7 Test list

The test list categories are given in the following subclauses.

7.1 TC Transaction sublayer test specification

7.1.1 Guidance on performing transaction sublayer tests

For each test, the expected message sequence, a test description and a check table for Information Elements (IE) within messages are given.

In the expected message sequence, primitives are shown at SP A [Implementation Under Test (IUT) side] only.

The function of the check table is to provide the contents of both the initiating message and the expected results in order to perform the checks in the test descriptions. The check table for IE within messages does not include information on the Component Portion or the User Abort Information IE contents, which are dependent on a specific application. In the check tables, messages from the IUT are described using the short form for any IE length, except for 1.1.3.1.1 which tests the length variations. However different forms complying with 3.3/Q.773 may be used in any test.

In order to test for pre and post test results such as the state machines being in the idle state, the following procedure is suggested:

Send a Continue to the IUT with the identical destination transaction ID (of a transaction that should be idle) and expect an Abort with unrecognized transaction ID cause value. This indicates that the state machine is in the idle state. If the Continue is accepted by the IUT and it gives a correct response, it was in the initiation sent state.

NOTE - The details of these confirmation tests are implementation dependant.

7.1.2 Transaction sublayer tests

No Details Available

For Further Study

NDA

FFS

*	* Validation and Compatibility					
All o	ther tests	s are Valid	dation Only			
1	Transa	action sub	layer			
	1.1	Valid fi	unction			
		1.1.1	Unstructi	ıred Dialogu	e	
*			1.1.1.1	Tested side	e sending	
*			1.1.1.2	Tested side	e receiving	
		1.1.2	Structure	d Dialogue		
			1.1.2.1	Clearing b	efore subseque	ent Message
				1.1.2.1.1	Valid clearing	ng from initiating side
*					1) Prearra	nged ending
*					2) Abort b	y the TR-User
				1.1.2.1.2	Valid clearing	ng from responding side
					1.1.2.1.2.1	IUT Sending
*						1) Basic ending
*						2) Prearranged ending
*						3) Abort by the TR-User
					1.1.2.1.2.2	IUT Receiving
*						1) Abort by the TR-User

Basic ending

Abort by Transaction Sublayer

2)

3)

			1.1.2.2	Clearing a	fter Continue I	Messa	age
				1.1.2.2.1	Valid clearing from initiating side		
					1.1.2.2.1.1	_	Γ Sending
*						1)	Basic ending
*						2)	Prearranged ending
*						3)	Abort by the TR-User
					1.1.2.2.1.2		Γ Receiving
*					1.11.2.2.11.2	1)	Basic ending
						2)	Abort by the Transaction Sublayer
*						3)	Abort by the TR-User
				1.1.2.2.2	Valid cleari		om responding side
				1.1.2.2.2	1.1.2.2.2.1	_	Γ Sending
*					1.1.2.2.2.1	1)	Basic ending
*						2)	Prearranged ending
*						3)	Abort by the TR-User
					1.1.2.2.2.2		Γ Receiving
*					1.1.2.2.2.2		
*						1)	Basic ending
*						2)	Abort by the Transaction Sublayer
*			1 1 2 2	CI :	6 G .: 1	3)	Abort by the TR-User
			1.1.2.3	_			age (Component Portion Not Present)
				1.1.2.3.1	Basic Endin	_	_
				1.1.2.3.2	Basic Endin	_	_
			1.1.2.4	_	_		saction Established
				1.1.2.4.1	IUT initiatin	_	
				1.1.2.4.2	IUT receiving	ng	
			1.1.2.5	TC Addres	•		
			- "	(For Furth	= :		
		1.1.3		g and value v			
			1.1.3.1	Encoding			
				1.1.3.1.1	Length varia		
					1.1.3.1.1.1		finite short
						1)	Component portion length in definite short form embedded in short form
						2)	Component portion length in definite short form embedded in long form
					1.1.3.1.1.2	De	finite long
						1)	Component portion length in definite long form embedded in long form
					1.1.3.1.1.3	Ind	efinite form
						1)	Component portion length in indefinite form embedded in indefinite form
			1.1.3.2	Value vari	ations		
				1.1.3.2.1	Transaction	ID	
*					1) Length	is on	e octet
*					-		ur octets
	1.2	Syntact	ically Inval	id Behaviour	, -		
	•	1.2.1	-		ormation eleme	ents	
			1.2.1.1	Begin Mes			
				_	0 length = 0		
				2) OTH	1 1 5		

2) OTID length > four octets

- 1.2.1.2 First Continue Message
 - 1) DTID length = 0
- 1.2.1.3 Subsequent Continue Message
 - 1) Component portion length incorrect
- 1.2.1.4 End Message
 - 1) DTID length > four octets
- 1.2.1.5 Abort Message
 - 1) Invalid P-Abort cause value
 - 2) P-Abort cause length incorrect
- 1.2.2 Invalid structure
 - 1.2.2.1 Unidirectional Message Type
 - 1) Unknown information element present
 - 1.2.2.2 Begin Message Type
 - 1) OTID absent
 - 2) Unknown information element present
 - 1.2.2.3 First Continue Message
 - 1) OTID absent
 - 2) DTID absent
 - 1) OTID duplicated
 - 2) DTID duplicated
 - 5) Unknown information element present
 - 1.2.2.4 Subsequent Continue Message
 - 1) OTID absent
 - 2) Unknown information element present
 - 1.2.2.5 End Message
 - 1) DTID absent
 - 1.2.2.6 Abort Message
 - 1) DTID absent
 - 1.2.2.7 Unknown Message
 - 1) OTID not included
 - 2) OTID included and DTID not included
 - 3) OTID included and DTID included
- 1.2.3 Invalid encoding (i.e. Rec. X.209 BER violation)
 - 1.2.3.1 Begin Message Type
 - 1) Invalid tag
 - 1.2.3.2 Continue Message Type
 - 1) Invalid tag
- 1.3 Incorporate Messages
 - 1.3.1 Continue Message Type
 - 1) Receipt of Continue message in Idle state with unassigned DTID
 - 1.3.2 End Message Type
 - 1) Receipt of End message in Idle state
 - 1.3.3 Abort Message Type
 - 1) Receipt of Abort message in Idle state
- 1.4 Multiple Transaction Encoding
 - 1.4.1 Valid Transaction Encoding
 - 1) New transaction request during transaction establishment
 - 2) New transaction request after transaction establishment
 - 1.4.2 Inopportune Messages
 - 1) Message with unassigned DTID during transaction establishment
 - 2) Message with unassigned DTID after transaction establishment

TEST NUMBER: 1.1.1.1	Sheet: 1 of 1		
REFERENCE: 3.3.3.1.1/Q.774			
TITLE: Valid Function; Unstructured l	Dialogue		
SUBTITLE: Tested side sending			
PURPOSE: To verify that signalling po	int A is able to correctly send a Unidirectional mo	essage	
PRE-TEST CONDITIONS: SP A (TSI	L) and SP B (TSL) are to be in the idle state		
CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPECTED MESSAGE SEQUENCE:			
SP A (TSL)		SP B (TSL)	
TR-UNI req.			
UNIDIRECTIONAL	>		
TEST DESCRIPTION			
Send a Unidirectional message f	from SP A to SP B.		
2. CHECK A: WAS THE UNIDI	RECTIONAL MESSAGE CORRECTLY SENT	FROM SP A?	
3. CHECK B: WAS THE TSL S' STATE AT SP A?	TATE MACHINE ASSOCIATED WITH THIS T	RANSACTION LEFT IN THE IDLE	
CHECK TABLE FOR INFORMATION	ELEMENTS WITHIN MESSAGES		
UNIDIRECTIONAL			
Message type tag: 01100001 Message type length: correct numb	er of octets		
Component portion tag: 01101100 Component portion length: correct number of octets			

TEST NUMBER: 1.1.1.2 Sheet: 1 of 1 REFERENCE: 3.3.3.1.2/Q.774 TITLE: Valid Function; Unstructured Dialogue SUBTITLE: Tested side receiving PURPOSE: To verify that signalling point A is able to correctly receive a Unidirectional message PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state TYPE OF TEST: VAT and CPT CONFIGURATION: 1 TYPE OF SP: SP EXPECTED MESSAGE SEQUENCE: SP A (TSL) SP B (TSL) UNIDIRECTIONAL TR-UNI ind. TEST DESCRIPTION Send a Unidirectional message from SP B to SP A. 1. CHECK A: WAS THE UNIDIRECTIONAL MESSAGE CORRECTLY RECEIVED AT SP A? 3. CHECK B: WAS THE TSL STATE MACHINE ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES UNIDIRECTIONAL Message type tag: 01100001 Message type length: correct number of octets Component portion tag: 01101100 Component portion length: correct number of octets

TEST NUMBER: 1.1.2.1.1 1)	Sheet: 1 of 1
---------------------------	---------------

REFERENCE: 3.3.3.2.1/Q.774 and 3.3.3.2.3/Q.774

TITLE: Valid Function; Structured Dialogue

SUBTITLE: Clearing before subsequent Message; Valid clearing from initiating side; Prearranged ending

PURPOSE: To verify that signalling point A is able to correctly send a Begin message and then terminate the transaction locally

by the "prearranged end" method

PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state

CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP

EXPECTED MESSAGE SEQUENCE:

SP A (TSL)

SP B (TSL)

TR-BEGIN req.

BEGIN

TR-END req.

(Prearranged)

TEST DESCRIPTION

- 1. Send a Begin message from SP A to SP B.
- 2. Before a reply is received from SP B, arrange for a TR-END request primitive (prearranged) to be passed to the TSL at
- CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? 3.
- CHECK B: VERIFY THAT AN END MESSAGE WAS NOT SENT BY SP A? 4.
- CHECK C: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

TEST NUMBER: 1.1.2.1.1 2) Sheet: 1 of 1

REFERENCE: 3.3.3.2.1/Q.774 and 3.3.3.2.4/Q.774

TITLE: Valid Function; Structured Dialogue

SUBTITLE: Clearing before subsequent Message; Valid clearing from initiating side; Abort by the TR-User

PURPOSE: To verify that signalling point A is able to correctly generate a Begin message and then terminate the transaction

locally by the "abort" method

PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state

CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP

EXPECTED MESSAGE SEQUENCE:

SP A (TSL) SP B (TSL)

TR-BEGIN req.

BEGIN -----

TR-U-ABORT req.

TEST DESCRIPTION

- 1. Send a Begin message from SP A to SP B.
- 2. Before a reply is received from SPB, arrange for a TR-U-ABORT request primitive to be passed to the TSL at SPA.
- 3. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?
- 4. CHECK B: WAS THE TR-U-ABORT REQUEST PURELY LOCAL AT SP A?
- 5. CHECK C: VERIFY THAT NO ABORT MESSAGE WAS SENT FROM SP A?
- 6. CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

TEST NUMBER: 1.1.2.1.2.1 1)			Sheet: 1 of 2
REFE	RENCE: 3.3.3.2.1/Q.774 and 3.3.	3.2.3/Q.774	
TITL	E: Valid Function; Structured Dial	ogue	
SUBT	TITLE: Clearing before subsequen	t Message; Valid clearing from responding side;	IUT Sending; Basic ending
PURF	POSE: To verify that signalling point end" method	int A is able to receive a Begin message and the	en terminate the transaction by the "basic
PRE-	TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP
EXPE	CCTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
		<	BEGIN
	TR-BEGIN ind.		
	TR-END req.		
	======> (Basic)		
	END	>	
	END		
TEST	DESCRIPTION		
TEST			
	DESCRIPTION Send a Begin message from SP B		o be passed to the TSL at SP A.
1.	DESCRIPTION Send a Begin message from SP B On receipt of BEGIN indication a	to SP A.	-
1.	DESCRIPTION Send a Begin message from SP B On receipt of BEGIN indication a CHECK A: WAS THE BEGIN	to SP A. arrange for a TR-END request primitive (basic) to	-
1. 2. 3.	DESCRIPTION Send a Begin message from SP B On receipt of BEGIN indication a CHECK A: WAS THE BEGIN CHECK B: WAS AN END ME	to SP A. arrange for a TR-END request primitive (basic) to MESSAGE CORRECTLY RECEIVED AT SP	A AND PASSED TO THE TR-USER?

TEST NUMBER: 1.1.2.1.2.1 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100 Component portion length: correct number of octets

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)
(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST NUMBER: 1.1.2.1.2.1 2) Sheet: 1 of 1

REFERENCE: 3.3.3.2.1/Q.774 and 3.3.3.2.3/Q.774

TITLE: Valid Function; Structured Dialogue

SUBTITLE: Clearing before subsequent Message; Valid clearing from responding side; IUT Sending; Prearranged ending

PURPOSE: To verify that the signalling point A is able to receive a Begin message and then terminate the transaction by the

"prearranged end" method

PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state

CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP

EXPECTED MESSAGE SEQUENCE:

SP A (TSL) SP B (TSL)

<----- BEGIN

TR-BEGIN ind.

TR-END req.
(Prearranged)

TEST DESCRIPTION

- 1. Send a Begin message from SP B to SP A.
- 2. On receipt of the BEGIN indication arrange for a TR-END request primitive (prearranged) to be passed to the TSL at SP A.
- 3. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A?
- 4. CHECK B: VERIFY THAT AN END MESSAGE WAS NOT SENT BY SP A?
- 5. CHECK C: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets

Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

TEST NUMBER: 1.1.2.1.2.1 3) Sheet: 1 of 1 REFERENCE: 3.3.3.2.1/Q.774 and 3.3.3.2.4/Q.774 TITLE: Valid Function; Structured Dialogue SUBTITLE: Clearing before subsequent Message; Valid clearing from responding side; IUT Sending; Abort by the TR-User PURPOSE: To verify that the signalling point A is able to receive a Begin message and then terminate the transaction by the "abort" method PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP EXPECTED MESSAGE SEQUENCE: SP A (TSL) SP B (TSL) **BEGIN** TR-BEGIN ind. TR-U-ABORT req. ABORT (U) TEST DESCRIPTION 1. Send a Begin message from SP B to SP A. 2. On receipt of the BEGIN indication arrange for a TR-U-ABORT request primitive to be passed to the TSL at SP A. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A? 3. CHECK B: WAS AN ABORT MESSAGE CORRECTLY SENT BY SP A? 4. CHECK C: WAS THE DTID IN THE ABORT MESSAGE THE SAME AS THE OTID IN THE BEGIN MESSAGE? 5. CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (U)

Message type tag: 01100111

Message type length: correct number of octets Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

User abort information tag: 01101011

User abort information length: correct number of octets

TEST NUMBER: 1.1.2.1.2.2 1) Sheet: 1 of 1 REFERENCE: 3.3.3.2.1/Q.774 and 3.3.3.2.4/Q.774 TITLE: Valid Function; Structured Dialogue SUBTITLE: Clearing before subsequent Message; Valid clearing from responding side; IUT Receiving; Abort by the TR-User PURPOSE: To verify that the signalling point A is able to terminate a transaction on reception of an Abort (U) message PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP EXPECTED MESSAGE SEQUENCE: SP A (TSL) SP B (TSL) TR-BEGIN req. BEGIN ABORT (U) TR-U-ABORT ind. TEST DESCRIPTION Send a Begin message from SP A to SP B. 1. 2. Arrange for SP B to send an U-Abort message to SP A. 3. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? 4. CHECK B: WAS THE ABORT MESSAGE CORRECTLY RECEIVED AT SP A? CHECK C: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE 5. STATE AT SP A? CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) Component portion tag: 01101100 Component portion length: correct number of octets ABORT (U) Message type tag: 01100111 Message type length: correct number of octets Destination transaction ID tag: 01001001 Destination transaction ID length: correct number of octets Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message) User abort information tag: 01101011 User abort information length: correct number of octets

TEST NUMBER: 1.1.2.1.2.2 2) Sheet: 1 of 1 REFERENCE: 3.3.3.2.1/Q.774 and 3.3.4/Q.774 TITLE: Valid Function; Structured Dialogue SUBTITLE: Clearing before subsequent Message; Valid clearing from responding side; IUT Receiving; Abort by Transaction PURPOSE: To verify that the signalling point A is able to terminate a transaction on reception of an Abort (P) message PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP EXPECTED MESSAGE SEQUENCE: SP B (TSL) SP A (TSL) TR-BEGIN req. BEGIN ABORT (P) TR-P-ABORT ind. TEST DESCRIPTION 1. Send a Begin message from SP A to SP B. 2. Arrange for SP B to send an P-Abort message to SP A. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? 3. 4. CHECK B: WAS THE ABORT MESSAGE CORRECTLY RECEIVED AT SP A? 5. CHECK C: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long) Component portion tag: 01101100 Component portion length: correct number of octets ABORT (P) Message type tag: 01100111 Message type length: correct number of octets Destination transaction ID tag: 01001001 Destination transaction ID length: correct number of octets Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message) P-Abort cause tag: 01001010 P-Abort cause length: one octet P-Abort cause value: INTEGER (between 0 and 4)

TEST N	IUMBER: 1.1.2.1.2.2 3)		Sheet: 1 of 1		
REFER	ENCE: 3.3.3.2.1/Q.774 and 3.3.	3.2.3/Q.774			
TITLE:	Valid Function; Structured Dial	ogue			
SUBTIT	ΓLE: Clearing before subsequent	Message; Valid clearing from responding side;	IUT Receiving; Basic ending		
PURPO	SE: To verify that the signalling	point A is able to terminate a transaction on rece	eption of an END message		
PRE-TE	EST CONDITIONS: SP A (TSL)	and SP B (TSL) are to be in the idle state			
(CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP		
EXPEC'	TED MESSAGE SEQUENCE:				
	SP A (TSL)		SP B (TSL)		
	TR-BEGIN req. ====>				
	BEGIN	>			
		<	END		
	TR-END ind. <======				
TEST D	DESCRIPTION				
1.	Send a Begin message from SP A	to SP B.			
2.	Arrange for SP B to send an End	message to SP A.			
3.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY SENT FROM SP A	?		
4.	CHECK B: WAS THE ABORT	MESSAGE CORRECTLY RECEIVED AT SP	A?		
5.	CHECK C: WERE TSL STAT STATE AT SP A?	E MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE		
СНЕСК	TABLE FOR INFORMATION	ELEMENTS WITHIN MESSAGES			
BEGIN					
Mes Mes	ssage type tag: 01100010 ssage type length: correct numbe	r of octets			
Orig	ginating transaction ID tag: 0100 ginating transaction ID length: coginating transaction ID value: OG	orrect number of octets			
	Component portion tag: 01101100 Component portion length: correct number of octets				
END					
	ssage type tag: 01100100 ssage type length: correct numbe	r of octets			
Dest	tination transaction ID tag: 0100 tination transaction ID length: co tination transaction ID value: (1001 orrect number of octets OCTET STRING (1-4 octets long) OTID value received in BEGIN message)			
	Component portion tag: 01101100 Component portion length: correct number of octets				

TEST NUMBER: 1.1.2.2.1.1 1)			Sheet: 1 of 2			
REFE	REFERENCE: 3.3.3.2.1/Q.774, 3.3.3.2.2/Q.774 and 3.3.3.2.3/Q.774					
TITLI	E: Valid Function; Structured Dial	ogue				
SUBT	TITLE: Clearing after Continue Me	essage;Valid clearing from initiating side, IUT Se	ending, Basic ending			
PURP	OSE: To verify that the signalling	point A is able to terminate the transaction by the	ne "basic end" method			
PRE-7	PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. SP B to respond with a Continue message on receipt of the Begin message					
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP			
EXPE	CTED MESSAGE SEQUENCE:					
	SP A (TSL)		SP B (TSL)			
	TR-BEGIN req.					
	BEGIN	>				
		<	CONTINUE			
	TR-CONTINUE ind.					
	TR-END req.					
	(Basic)					
	END	>				
TFST	DESCRIPTION					
1.		to SP B. Arrange for SP B to respond with a Co	untinue message			
2.	Send a Begin message from SP A to SP B. Arrange for SP B to respond with a Continue message					
3.	On receipt of the CONTINUE indication arrange for a TR-END request primitive (basic) to be passed to the TSL at SP A.					
4.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY RECEIVED BY THE TSL AT SP A?					
5.		ESSAGE CORRECTLY SENT BY SP A?				
6.		N THE END MESSAGE THE SAME AS THE	OTID IN THE CONTINUE MESSAGE?			
		TE MACHINES ASSOCIATED WITH THIS				
7.	STATE AT SP A?	E MACHINES ASSOCIATED WITH THIS	INANSACTION LEFT IN THE IDLE			

TEST NUMBER: 1.1.2.2.1.1 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

Component portion tag: 01101100

TEST NUMBER: 1.1.2.2.1.1 2)			Sheet: 1 of 2			
REFE	REFERENCE: 3.3.3.2.1/Q.774, 3.3.3.2.2/Q.774 and 3.3.3.2.3/Q.774					
TITLI	E: Valid Function; Structured Dial	ogue				
SUBT	TITLE: Clearing after Continue Me	essage;Valid clearing from initiating side, IUT Se	ending, Prearranged ending			
PURP	OSE: To verify that signalling poin	nt A is able to terminate the transaction by the "p	orearranged end" method			
PRE-	PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. SP B to respond with a Continue message on receipt of the Begin message					
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP			
EXPE	CTED MESSAGE SEQUENCE:					
	SP A (TSL)		SP B (TSL)			
	TR-BEGIN req.					
	BEGIN	>				
		<	CONTINUE			
	TR-CONTINUE ind.					
	TR-END req.					
	(Prearranged)					
TEST	DESCRIPTION					
1.	Send a Begin message from SP A	to SPB. Arrange for SPB to respond with a Co	ontinue message			
2.	On receipt of the CONTINUE indication arrange for a TR-END request primitive (prearranged) to be passed to the TSL at SP A.					
3.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY SENT FROM SP A	?			
4.	CHECK B: WAS THE CONTIN	NUE MESSAGE CORRECTLY RECEIVED A	Γ SP A?			
5.		HE TR-END REQUEST PRIMITIVE WAS PU IOT GENERATED AND SENT BY SP A?	JRELY LOCAL AND THAT AN END			
6.	CHECK D: WERE TSL STAT STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE			

TEST NUMBER: 1.1.2.2.1.1 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST NUMBER: 1.1.2.2.1.1 3)			Sheet: 1 of 2		
REFE	REFERENCE: 3.3.3.2.1/Q.774, 3.3.3.2.2/Q.774 and 3.3.3.2.4/Q.774				
TITLI	E: Valid Function; Structured Dial	ogue			
SUBT	TITLE: Clearing after Continue M	essage;Valid clearing from initiating side, IUT S	ending, Abort by the TR-User		
PURP	POSE: To verify that the signalling	point A is able to terminate the transaction by th	e "abort" method		
PRE-	TEST CONDITIONS: SP A (TSL receipt of t) and SP B (TSL) are to be in the idle state. SP I he Begin message	B to respond with a Continue message on		
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP		
EXPE	CCTED MESSAGE SEQUENCE:				
	SP A (TSL)		SP B (TSL)		
	TR-BEGIN req.				
	BEGIN	>			
		<	CONTINUE		
	TR-CONTINUE ind.				
	TR II APORT was				
	TR-U-ABORT req. =====>				
	ABORT (U)	>			
TEST	DESCRIPTION				
1.	1. Send a Begin message from SP A to SP B. Arrange for SP B to respond with a Continue message				
2.	On receipt of the CONTINUE indication arrange for a TR-U-ABORT request primitive to be passed to TSL at SP A.				
3.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?				
4.	CHECK B: WAS THE CONTI	NUE MESSAGE CORRECTLY RECEIVED A	ΓSP A?		
5.	CHECK C: WAS THE ABORT	MESSAGE CORRECTLY SENT BY SP A?			
6.	CHECK D: WAS THE DTID MESSAGE?	IN THE ABORT MESSAGE THE SAME	AS THE OTID IN THE CONTINUE		
7.	CHECK E: WERE TSL STAT STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE		

TEST NUMBER: 1.1.2.2.1.1 3) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

User abort information tag: 01101011

User abort information length: correct number of octets

TEST	NUMBER: 1.1.2.2.1.2 1)		Sheet: 1 of 2		
REFE	RENCE: 3.3.3.2.1.2/Q.774, 3.3.3.2	2.2/Q.774 and 3.3.3.2.3/Q.774			
TITLI	E: Valid Function; Structured Dialo	egue			
SUBT	TTLE: Clearing after Continue Me	ssage;Valid clearing from initiating side, IUT R	eceiving, Basic ending		
PURP	OSE: To verify that signalling po reception of an End message	int A is able to generate a Continue message	e and then terminate the transaction on		
PRE-7	TEST CONDITIONS: SP A (TSL)	and SP B (TSL) are to be in the idle state			
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP		
EXPE	CTED MESSAGE SEQUENCE:				
	SP A (TSL)		SP B (TSL)		
		<	BEGIN		
	TR-BEGIN ind.				
	TR-CONTINUE req.				
	======================================				
	CONTINUE	>			
		<	END		
	TR-END ind. <========				
TEST	TEST DESCRIPTION				
1.	Arrange for SP B to send a Begin	message to SP A.			
2.	Arrange for SP A to respond with	a Continue message.			
3.	Arrange for SP B to respond with an End message.				
4.	CHECK A: WAS THE BEGIN I	MESSAGE CORRECTLY RECEIVED AT SP	A?		
5.	CHECK B: WAS THE CONTIN	IUE MESSAGE CORRECTLY SENT FROM S	SP A?		
6.	CHECK C: WAS THE END ME	ESSAGE CORRECTLY RECEIVED AT SP A?			
7.	CHECK D: WERE TSL STATE STATE AT SP A?	E MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE		

TEST NUMBER: 1.1.2.2.1.2 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

Component portion tag: 01101100

TEST	NUMBER: 1.1.2.2.1.2 2)		Sheet: 1 of 2		
REFE	RENCE: 3.3.3.2.1.2/Q.774 and 3.3	3.4/Q.774			
TITLI	E: Valid Function; Structured Dialo	ogue			
SUBT	TTLE: Clearing after Continue Me	ssage; Valid clearing from initiating side, IUT R	eceiving, Abort by theTSL		
PURP	OSE: To verify that signalling poreception of an Abort message	oint A is able to generate a Continue message ge by the peer TSL	e and then terminate the transaction on		
PRE-	TEST CONDITIONS: SP A (TSL)	and SP B (TSL) are to be in the idle state			
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP		
EXPE	CTED MESSAGE SEQUENCE:				
	SP A (TSL)		SP B (TSL)		
		<	BEGIN		
	TR-BEGIN ind.				
	TR-CONTINUE req.				
	======>				
	CONTINUE	>			
		<	ABORT (P)		
	TR-P-ABORT ind. <======				
TEST	DESCRIPTION				
1.	Arrange for SP B to send a Begin	message to SP A.			
2.	Arrange for SP A to respond with	a Continue message.			
3.	Arrange for SP B to respond with an Abort (P) message.				
4.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY RECEIVED AT SP	A?		
5.	CHECK B: WAS THE CONTIN	NUE MESSAGE CORRECTLY SENT FROM S	SP A?		
6.	CHECK C: WAS THE ABORT	MESSAGE CORRECTLY RECEIVED AT SP	A?		
7.	CHECK D: WERE TSL STAT STATE AT SP A?	E MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE		

TEST NUMBER: 1.1.2.2.1.2 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

P-Abort cause tag: 01001010 P-Abort cause length: one octet P-Abort cause value: INTEGER (0 ... 4)

Recommendation Q.787

(03/93)

TEST	NUMBER: 1.1.2.2.1.2 3)	Sheet: 1 of 2				
REFERENCE: 3.3.3.2.1.2/Q.774 and 3.3.3.2.4/Q.774						
TITLI	E: Valid Function; Structured Dialo	ogue				
SUBT	TTLE: Clearing after Continue Me	essage; Valid clearing from initiating side, IUT R	eceiving, Abort by the TR-User			
PURPOSE: To verify that signalling point A is able to generate a Continue message and then terminate the transaction on reception of an Abort message by the peer TR-User						
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state						
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP			
EXPE	CTED MESSAGE SEQUENCE:					
	SP A (TSL)		SP B (TSL)			
		<	BEGIN			
	TR-BEGIN ind.					
	TR-CONTINUE req.					
	======>					
	CONTINUE	>				
		<	ABORT (U)			
	TR-U-ABORT ind. <=======					
TEST	DESCRIPTION					
1.	Arrange for SP B to send a Begin message to SP A.					
2.	Arrange for SP A to respond with a Continue message.					
3.	Arrange for SP B to respond with an Abort (U) message.					
4.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A?					
5.	CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY SENT FROM SP A?					
6.	CHECK C: WAS THE ABORT MESSAGE CORRECTLY RECEIVED AT SP A?					
7.	CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?					

TEST NUMBER: 1.1.2.2.1.2 3) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (U)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

User abort information tag: 01101011

User abort information length: correct number of octets

TEST	NUMBER: 1.1.2.2.2.1 1)	Sheet: 1 of 2				
REFERENCE: 3.3.3.2.1/Q.774, 3.3.3.2.2/Q.774 and 3.3.3.2.3/Q.774						
TITLE: Valid Function; Structured Dialogue						
SUBTITLE: Clearing after Continue Message; Valid clearing from responding side, IUT Sending, Basic ending						
PURPOSE: To verify that signalling point A is able to generate a Continue message and then terminate the transaction by the "basic end" method						
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state						
	CONFIGURATION: 1 TYPE OF TEST: VAT and CPT	TYPE OF SP: SP				
EXPE	CTED MESSAGE SEQUENCE:					
	SP A (TSL)	SP B (TSL)				
	<	BEGIN				
	TR-BEGIN ind.					
	TR-CONTINUE req.					
	=======>					
	CONTINUE>					
	TR-END req. =====>					
	(Basic)					
	END>					
TEST	DESCRIPTION					
1.	Arrange for SP B to send a Begin message to SP A.					
2.	Arrange for SP A to respond with a Continue message.					
3.	Terminate the transaction with an End (Basic) message from SP A.					
4.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A?					
5.	CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY SENT BY THE TSL AT SP A?					
6.	CHECK C: WAS AN END MESSAGE CORRECTLY SENT BY SP A?					
7.	CHECK D: WAS THE DTID IN THE CONTINUE AND END MESSAGES THE SAME AS THE OTID IN THE BEGIN MESSAGE?					
8.	CHECK E: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?					

TEST NUMBER: 1.1.2.2.2.1 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

Component portion tag: 01101100

TEST	NUMBER: 1.1.2.2.2.1 2)	Sheet: 1 of 2				
REFERENCE: 3.3.3.2.1/Q.774, 3.3.3.2.2/Q.774 and 3.3.3.2.3/Q.774						
TITLE: Valid Function; Structured Dialogue						
SUBT	TITLE: Clearing after Continue Message; Valid clearing from responding side,	IUT Sending, Prearranged ending				
PURPOSE: To verify that signalling point A is able to generate a Continue message and then terminate the transaction by the "prearranged end" method						
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state						
	CONFIGURATION: 1 TYPE OF TEST: VAT and CPT	TYPE OF SP: SP				
EXPE	CCTED MESSAGE SEQUENCE:					
	SP A (TSL)	SP B (TSL)				
	<	BEGIN				
	TR-BEGIN ind.					
TR-CONTINUE req.						
	CONTINUE	->				
	TR-END req.					
	(Prearranged)					
TEST	DESCRIPTION					
1.	Arrange for SP B to send a Begin message to SP A.					
2.	Arrange for SP A to respond with a Continue message.					
3.	Terminate the transaction with a TR-END request primitive (prearranged) from SP A.					
4.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A?					
5.	CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY SENT BY THE TSL AT SP A?					
6.	CHECK C: VERIFY THAT THE TR-END REQUEST PRIMITIVE WAS PURELY LOCAL AND THAT AN END MESSAGE WAS NOT GENERATED AND SENT BY SP A?					
7.	CHECK D: WAS THE DTID IN THE CONTINUE MESSAGE THE SAME AS THE OTID IN THE BEGIN?					
8.	CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?					

TEST NUMBER: 1.1.2.2.2.1 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST NUMBER: 1.1.2.2.2.1 3)			Sheet: 1 of 2			
REFERENCE: 3.3.3.2.1/Q.774, 3.3.3.2.2/Q.774 and 3.3.3.2.4/Q.774						
TITLE: Valid Function; Structured Dialogue						
SUBTITLE: Clearing after Continue Message; Valid clearing from initiating side, IUT Sending, Abort by the TR-User						
PURPOSE: To verify that signalling point A is able to generate a Continue message and then terminate the transaction by the "abort" method						
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state						
	CONFIGURATION: 1	TYPE OF TEST: VAT and CP	TYPE OF SP: SP			
EXPE	ECTED MESSAGE SEQUENCE:					
	SP A (TSL)		SP B (TSL)			
		<	BEGIN			
	TR-BEGIN ind.					
	<=======					
	TR-CONTINUE req. =====>					
	CONTINUE	>				
	TR-U-ABORT req.					
	ABORT (U)	>				
	(0)					
TEST	DESCRIPTION					
1.	Arrange for a Begin message to be sent from SP B to SP A.					
2.	Arrange for SP A to respond with a Continue message, then abort the transaction by passing a TR-U-ABORT request primitive to the TSL at SP B.					
3.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A?					
4.	CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY SENT FROM SP A?					
5.	CHECK C: WAS THE ABORT MESSAGE CORRECTLY SENT FROM SP A?					
6.	CHECK D: WAS THE DTID IN THE CONTINUE AND ABORT MESSAGES THE SAME AS THE OTID IN THE BEGIN MESSAGE?					
7.	CHECK E: WERE TSL STAT STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE			

TEST NUMBER: 1.1.2.2.2.1 3) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

User abort information tag: 01101011

User abort information length: correct number of octets

TEST	NUMBER: 1.1.2.2.2.2 1)		Sheet: 1 of 2
REFE	REFERENCE: 3.3.3.2.1/Q.774 and 3.3.3.2.3/Q.774		
TITLI	E: Valid Function; Structured dialo	ogue	
SUBT	TTLE: Clearing after Continue me	ssage; Valid clearing from responding side; IUT	receiving; Basic ending
PURP	OSE: To verify that the signalling Continue message	point A is able to terminate the transaction on i	reception of an End message following a
PRE-T	TEST CONDITIONS: SP A (TSL)	and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
		<	CONTINUE
	TR-CONTINUE ind.		
	<=======		
< END TR-END ind.			
	1 R-END ind. <====================================		
TEST	DESCRIPTION		
1.	Arrange for SP A to send a Begin	message to SP B.	
2.	Arrange for SP B to respond with	a Continue message.	
3.	Terminate the transaction with an End (basic) message from SP B.		
4.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY SENT FROM SP A?	,
5.	CHECK B: WAS THE CONTIL	NUE MESSAGE CORRECTLY RECEIVED AT	Γ SP A?
6.	CHECK C: WAS THE END M	ESSAGE CORRECTLY RECEIVED AT SP A?	,
7.	CHECK D: WERE TSL STAT STATE AT SP A?	E MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE

TEST NUMBER: 1.1.2.2.2.2 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST	NUMBER: 1.1.2.2.2.2 2)		Sheet: 1 of 2	
REFE	REFERENCE: 3.3.3.2.1/Q.774 and 3.3.4/Q.774			
TITLI	E: Valid function; Structured dialo	gue		
SUBT	TTLE: Clearing after Continue me	ssage; Valid clearing from initiating side; IUT re	eceiving; Abort by the TSL	
PURP	OSE: To verify that the signallin following a Continue message	g point A is able to terminate the transaction	on reception of an Abort (P) message	
PRE-T	TEST CONDITIONS: SP A (TSL)	and SP B (TSL) are to be in the idle state		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE SEQUENCE:			
	SP A (TSL)		SP B (TSL)	
	TR-BEGIN req.			
	BEGIN	>		
		<	CONTINUE	
	TR-CONTINUE ind.			
	\	<	ABORT (P)	
TR-P-ABORT ind.				
	<=====================================			
TEST	DESCRIPTION			
1.	Arrange for SP A to send a Begin	message to SP B.		
2.	Arrange for SP B to respond with	a Continue message.		
3.	Terminate the transaction with an Abort (P) message from SP B.			
4.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?		,	
5.	CHECK B: WAS THE CONTIL	NUE MESSAGE CORRECTLY RECEIVED A	ΓSP A?	
6.	CHECK C: WAS THE ABORT	MESSAGE CORRECTLY RECEIVED AT SP	A?	
7.	CHECK D: WERE TSL STAT STATE AT SP A?	E MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE	

TEST NUMBER: 1.1.2.2.2.2 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

P-Abort cause tag: 01001010 P-Abort cause length: one octet P-Abort cause value: INTEGER (0 .. 4)

TEST	NUMBER: 1.1.2.2.2.2 3)		Sheet: 1 of 2
REFE	REFERENCE: 3.3.3.2.1/Q.774 and 3.3.3.2.4/Q.774		
TITLI	E: Valid function; Structured dialo	gue	
SUBT	TTLE: Clearing after Continue me	ssage; Valid clearing from responding side; IUT	receiving; Abort by theTR-User
PURP	OSE: To verify that the signallin following a Continue message	g point A is able to terminate the transaction ge	on reception of an Abort (U) message
PRE-	TEST CONDITIONS: SP A (TSL)	and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
		<	CONTINUE
	TR-CONTINUE ind.		
		<	ABORT (U)
TR-U-ABORT ind.			
TEST	DESCRIPTION		
1.	Arrange for SP A to send a Begin	-	
2.	Arrange for SP B to respond with	-	
3.	Terminate the transaction with an Abort (U) message from SP B.		
4.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY SENT FROM SP A	
5.	CHECK B: WAS THE CONTIN	NUE MESSAGE CORRECTLY RECEIVED A	Γ SP A?
6.	CHECK C: WAS THE ABORT	MESSAGE CORRECTLY RECEIVED AT SP	A?
7.	CHECK D: WERE TSL STAT STATE AT SP A?	E MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE

TEST NUMBER: 1.1.2.2.2.2 3) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (U)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

User abort information tag: 01101011

User abort information length: correct number of octets

TEST	NUMBER: 1.1.2.3.1		Sheet: 1 of 2
REFERENCE: Q.774			
TITLI	E: Valid function; Structured dialo	gue	
SUBT	TTLE: Clearing after Continue me	essage (component portion not present); Basic en	nding IUT sending
PRE-7	TEST CONDITIONS: SP A (TSL)) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
		<	CONTINUE
	TR-CONTINUE ind.		
	TR-END req.		
	END	>	
TEST	DESCRIPTION		
1.	Arrange for SP A to send a Begin	message to SP B.	
2.	Arrange for SP B to send a Contin	nue message to SP A without CP.	
3.	Arrange for SP A to send an End	message to SP B	
4.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY SENT FROM SP A	?
5.	CHECK B: WAS THE CONTI	NUE MESSAGE CORRECTLY RECEIVED A	AT SP A?
6.	CHECK C: WAS THE END M	ESSAGE CORRECTLY SENT FROM SP A?	
7.	CHECK D: WAS THE TSL ST	ATE MACHINE LEFT IN THE IDLE STATE	AT SP A?

TEST NUMBER: 1.1.2.3.1 Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets (range 1-4)
Destination transaction ID value: OCTET STRING (1-4 octets long)
(OTID value received in BEGIN message)

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets (range 1-4) Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

Component portion tag: 01101100

TEST	ST NUMBER: 1.1.2.3.2		Sheet: 1 of 2	
REFE	REFERENCE: 3.2.1.3/Q.774			
TITL	E: Valid function; Structured dialo	gue		
SUBT	TITLE: Clearing after Continue me	essage (component portion not present); Basic en	ding IUT receiving	
PURF	POSE: To verify that SP A is able to	to accept a Begin message without CP		
PRE-	TEST CONDITIONS: SP A (TSL) and SP B (TSL) are in the idle state		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CCTED MESSAGE SEQUENCE:			
	SP A (TSL)		SP B (TSL)	
		<	BEGIN	
	TR-BEGIN ind.			
	TR-CONTINUE req.			
	======>			
		>	CONTINUE	
		<	END	
		-		
TR-END ind.				
TEST	DESCRIPTION			
1.	Arrange for SP B to send a BEGI	N message to SP A without CP.		
2.	Arrange for SP A to send a CON	TINUE messager to SP B.		
3.	Arrange for SP B to send an END	message to SP A without CP.		
4.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY RECEIVED AT SP	A?	
5.	CHECK B: WAS THE CONTI	NUE MESSAGE CORRECTLY SENT FROM S	SP A?	
6.	CHECK C: WAS THE END MESSAGE CORRECTLY RECEIVED AT SP A?			
7.	CHECK E: WAS THE TSL STATE MACHINE LEFT IN THE IDLE STATE AT SP A?			

TEST NUMBER: 1.1.2.3.2 Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4)
Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets (range 1-4)
Destination transaction ID value: OCTET STRING (1-4 octets long)
(OTID value received in BEGIN message)

(OTID Value received in BEOIT

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets (range 1-4) Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

TEST NUMBER: 1.1.2.4.1	TEST NUMBER: 1.1.2.4.1	
REFERENCE: 3.2.1.3/Q.774		
TITLE: Valid function; Structured dialo	ogue	
SUBTITLE: Message exchange after tr	ansaction Established; IUT initiating	
PURPOSE: To verify the correct messa	ge flow between SP A and SP B, after transaction	n established (IUT initiating)
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state	
CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP		
EXPECTED MESSAGE SEQUENCE:		
SP A (TSL)		SP B (TSL)
TR-BEGIN req.		
BEGIN	>	
	<	CONTINUE
TR-CONTINUE ind.		
TR-CONTINUE req.		
CONTINUE		
CONTINUE	<	END
TR-END ind.	\	END
<=====================================		
TEST DESCRIPTION		
1. Arrange for SP A to send a Begin	n message to SP B.	
2. Arrange for SP B to send a Conti	nue message to SP A.	
3. Arrange for SP A to send a Conti	inue message to SP B.	
4. Arrange for SP B to send an ENI	O message to SP A.	
5. CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY SENT FROM SP AS	?
6. CHECK B: WAS THE CONTI	NUE MESSAGE CORRECTLY RECEIVED A	Г SP А?
7. CHECK C: WAS THE CONTI	NUE MESSAGE CORRECTLY SENT FROM S	SP A?
8. CHECK D: WAS THE END M	IESSAGE CORRECTLY RECEIVED AT SP A	
9. CHECK E: WAS THE TSL ST	CHECK E: WAS THE TSL STATE MACHINE LEFT IN THE IDLE STATE AT SP A?	

TEST NUMBER: 1.1.2.4.1 Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: 0 (invalid length)

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: 0 (invalid length)

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100101

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: 0 (invalid length)

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST	TEST NUMBER: 1.1.2.4.2		Sheet: 1 of 2
REFE	RENCE: 3.2.1.3/Q.774		
TITLI	E: Valid function; Structured dialo	gue	
SUBT	TITLE: Message exchange after tra	nsaction established; IUT receiving	
PURP	OSE: To verify the correct message	ge flow between SP A and SP B, after transaction	n established (IUT receiving)
PRE-	TEST CONDITIONS: SP A (TSL)) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP		
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
		<	BEGIN
	TR-BEGIN ind.		
	TR-CONTINUE req.		
	CONTINUE	>	
		<	CONTINUE
	TR-CONTINUE ind.		
	TR-END req.		
	END	>	
TEST	DESCRIPTION		
1.	Arrange for SP B to send a Begin	message to SP A.	
2.	Arrange for SP A to send a Conti	nue message to SP B.	
3.	Arrange for SP B to send a Contin	nue message to SP A.	
4.	Arrange for SP A to send an ENI	message to SP B.	
5.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A?		A?
6.	CHECK B: WAS THE CONTI	NUE MESSAGE CORRECTLY SENT FROM S	SP A?
7.	CHECK C: WAS THE CONTI	NUE MESSAGE CORRECTLY RECEIVED A	Γ SP A?
8.	CHECK D: WAS THE END MESSAGE CORRECTLY SENT FROM SP A?		
9.	CHECK E: WAS THE TSL STATE MACHINE LEFT IN THE IDLE STATE AT SP A?		

TEST NUMBER: 1.1.2.4.2 Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: 0 (invalid length)

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets (range 1-4) Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets (range 1-4) Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100101

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets (range 1-4) Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)

Component portion tag: 01101100

TEST NUMBER: 1.1.2.5	Sheet: 1 of 1	
REFERENCE:		
TITLE: TC Adressing (For Further Study)		

TEST	TEST NUMBER: 1.1.3.1.1.1 1)		Sheet: 1 of 2	
REFE	ERENCE: 3.3/Q.774			
TITL	E: Valid function; Encoding and	value variations		
SUBT	FITLE: Encoding variations; Leng in short form	th variations; Definite short; Component portion	n length in definite short form embedded	
PURI	PURPOSE: To verify that signalling point A is able to accept a Begin message whose length is encoded using the definite short form and with a component portion whose length is encoded using the definite short form			
PRE-	TEST CONDITIONS: SP A (TSI	L) and SP B (TSL) are to be in the idle state		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPI	ECTED MESSAGE SEQUENCE:			
	SP A (TSL)		SP B (TSL)	
	< BEGIN			
	TR-BEGIN ind. <======			
	TR-END req.			
	=====> (Basic)			
	END	>		
TEST	DESCRIPTION			
1.	Arrange for SP B to send a Begi	n message to SP A with lengths encoded as descr	ibed in the purpose of the test.	
2.	2. Arrange for SP A to respond with an End message.			
3.	3. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A AND PASSED TO THE TR-USER?		A AND PASSED TO THE TR-USER?	
4.	CHECK B: WAS AN END M	ESSAGE CORRECTLY SENT BY SP A?		
5.	CHECK C: WAS THE DTID	N THE END MESSAGE THE SAME AS THE	OTID IN THE BEGIN MESSAGE?	
6.	CHECK D: WERE TSL STA STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE	

TEST NUMBER: 1.1.3.1.1.1 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets coded in definite short form

Originating transaction ID tag: 01001000 Originating transaction ID length: one octet Originating transaction ID value: OCTET STRING (1 octet)

Component portion tag: 01101100

Component portion length: correct number of octets coded in definite short form

END

Message type tag: 01100100

Message type length: correct number of octets Destination transaction ID tag: 01001001

Destination transaction ID length: one octet

Destination transaction ID value: OCTET STRING (1 octet)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST	TEST NUMBER: 1.1.3.1.1.1 2)		Sheet: 1 of 2
REFE	REFERENCE: 3.3/Q.774		
TITLE	E: Valid function; Encoding and va	alue variations	
SUBT	ITLE: Encoding variations; Lengt in long form	h variations; Definite short; Component portion	n length in definite short form embedded
PURP	OSE: To verify that signalling portion and with a component	int A is able to accept a Begin message whose portion whose length is encoded using the defini	length is encoded using the definite long ite short form
PRE-T	TEST CONDITIONS: SP A (TSL)) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
		<	BEGIN
	TR-BEGIN ind.		
	TR-END reg.		
	======> (Basic)		
	END	>	
TEST	DESCRIPTION		
1.	Arrange for SP B to send a Begin	message to SP A with lengths encoded as descri	ibed in the purpose of the test.
2.	Arrange for SP A to respond with	an End message.	
3.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY RECEIVED AT SP	A AND PASSED TO THE TR-USER?
4.	CHECK B: WAS AN END ME	SSAGE CORRECTLY SENT BY SP A?	
5.	CHECK C: WAS THE DTID IN	N THE END MESSAGE THE SAME AS THE C	OTID IN THE BEGIN MESSAGE?
6.	CHECK D: WERE TSL STAT	E MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE

TEST NUMBER: 1.1.3.1.1.1 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets coded in definite long form

Originating transaction ID tag: 01001000 Originating transaction ID length: one octet Originating transaction ID value: OCTET STRING (1 octet)

Component portion tag: 01101100

Component portion length: correct number of octets coded in definite short form

END

Message type tag: 01100100

Message type length: correct number of octets Destination transaction ID tag: 01001001

Destination transaction ID length: one octet

Destination transaction ID value: OCTET STRING (1 octet)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST	TEST NUMBER: 1.1.3.1.1.2 1)		Sheet: 1 of 2
REFE	RENCE: 3.3/Q.774		
TITLI	E: Valid function; Encoding and va	alue variations	
SUBT	TTLE: Encoding variations; Length long form	h variations; Definite long; Component portion	length in definite long form embedded in
PURP	POSE: To verify that signalling point form and with a component	int A is able to accept a Begin message whose portion whose length is encoded using the defini	length is encoded using the definite long te long form
PRE-7	TEST CONDITIONS: SP A (TSL)) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CCTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
		<	BEGIN
	TR-BEGIN ind.		
	TR-END reg.		
	======> (Basic)		
	END	>	
TEST	DESCRIPTION		
1.	Arrange for SP B to send a Begin	message to SP A with lengths encoded as descri	ibed in the purpose of the test.
2.	Arrange for SP A to respond with	an End message.	
3.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY RECEIVED AT SP	A AND PASSED TO THE TR-USER?
4.	CHECK B: WAS AN END ME	SSAGE CORRECTLY SENT BY SP A?	
5.	CHECK C: WAS THE DTID IN	N THE END MESSAGE THE SAME AS THE G	OTID IN THE BEGIN MESSAGE?
6.	CHECK D: WERE TSL STAT STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE

TEST NUMBER: 1.1.3.1.1.2 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets coded in definite long form

Originating transaction ID tag: 01001000 Originating transaction ID length: one octet Originating transaction ID value: OCTET STRING (1 octet)

Component portion tag: 01101100

Component portion length: correct number of octets coded in definite long form

END

Message type tag: 01100100

Message type length: correct number of octets Destination transaction ID tag: 01001001

Destination transaction ID length: one octet

Destination transaction ID value: OCTET STRING (1 octet)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST	NUMBER: 1.1.3.1.1.3 1)		Sheet: 1 of 2
REFE	REFERENCE: 3.3/Q.774		
TITLE	E: Valid function; Encoding and va	alue variations	
SUBT	TTLE: Encoding variations; Lengt indefinite form	h variations; Indefinite form; Component portion	on length in indefinite form embedded in
PURP	OSE: To verify that signalling poi and with a component portion	nt A is able to accept a Begin message whose le on whose length is encoded using the indefinite f	ngth is encoded using the indefinite form
PRE-T	TEST CONDITIONS: SP A (TSL)) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
		<	BEGIN
	TR-BEGIN ind.		
	TR-END reg.		
	======> (Basic)		
	END	>	
TEST	DESCRIPTION		
1.	Arrange for SP B to send a Begin	message to SP A with lengths encoded as descri	ibed in the purpose of the test.
2.	Arrange for SP A to respond with	an End message.	
3.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY RECEIVED AT SP	A AND PASSED TO THE TR-USER?
4.	CHECK B: WAS AN END ME	SSAGE CORRECTLY SENT BY SP A?	
5.	CHECK C: WAS THE DTID IN	N THE END MESSAGE THE SAME AS THE O	OTID IN THE BEGIN MESSAGE?
6.	CHECK D: WERE TSL STAT	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE

TEST NUMBER: 1.1.3.1.1.3 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets coded in indefinite form

Originating transaction ID tag: 01001000 Originating transaction ID length: one octet Originating transaction ID value: OCTET STRING (1 octet)

Component portion tag: 01101100

Component portion length: correct number of octets coded in indefinite form Component contents provided by TC user

EOC Tag: 00000000, Length: 00000000

END

Message type tag: 01100100

Message type length: correct number of octets Destination transaction ID tag: 01001001 Destination transaction ID length: one octet

Destination transaction ID value: OCTET STRING (1 octet)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST NUMBER: 1.1.3.2.1 1)		Sheet: 1 of 2	
REFERENCE: 5.3/Q.774			
TITLI	E: Valid function; Encoding and v	alue variations	
SUBT	TITLE: Value variations; Transacti	on ID; Length is one octet	
PURP	POSE: To verify that signalling poi	nt A is able to deal with correct encoding of OTI	D information element (1 octet long)
PRE-7	TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP
EXPE	ECTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
		<	BEGIN
	TR-BEGIN ind.		
	TR-END req.		
	=====> (Basic)		
	END	>	
TEST DESCRIPTION			
1.	Arrange for SP B to send a Begin message to SP A with an OTID 1 octet long.		
2.	Arrange for SP A to respond with an End message.		
3.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A AND PASSED TO THE TR-USER?		
4.	CHECK B: WAS AN END MESSAGE CORRECTLY SENT BY SP A?		
5.	CHECK C: WAS THE DTID IN THE END MESSAGE THE SAME AS THE OTID IN THE BEGIN MESSAGE?		
6.	CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?		

TEST NUMBER: 1.1.3.2.1 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000 Originating transaction ID length: one octet Originating transaction ID value: OCTET STRING (1 octet)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100100

Message type length: correct number of octets Destination transaction ID tag: 01001001

Destination transaction ID length: one octet
Destination transaction ID value: OCTET STRING (1 octet)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST NUMBER: 1.1.3.2.1 2)		Sheet: 1 of 2	
REFERENCE: 5.3/Q.774			
TITLI	E: Valid function; Encoding and va	alue variations	
SUBT	TITLE: Value variations; Transaction	on ID; Length is four octets	
PURP	OSE: To verify that signalling poin	nt A is able to deal with correct encoding of OTI	D information element (4 octets long)
PRE-7	TEST CONDITIONS: SP A (TSL)) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
		<	BEGIN
	TR-BEGIN ind.		
	TR-END req.		
	=====> (Basic)		
	END	>	
TEST DESCRIPTION			
1.	Arrange for SP B to send a Begin message to SP A with an OTID four octets long.		
2.	Arrange for SP A to respond with an End message.		
3.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A AND PASSED TO THE TR-USER?		
4.	CHECK B: WAS AN END MESSAGE CORRECTLY SENT BY SP A?		
5.	CHECK C: WAS THE DTID IN THE END MESSAGE THE SAME AS THE OTID IN THE BEGIN MESSAGE?		
6.	CHECK D: WERE TSL STAT STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE

TEST NUMBER: 1.1.3.2.1 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000 Originating transaction ID length: four octets Originating transaction ID value: OCTET STRING (4 octets)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100100

Message type length: correct number of octets Destination transaction ID tag: 01001001

Destination transaction ID length: four octets
Destination transaction ID value: OCTET STRING (4 octets)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST NUMBER: 1.2.1.1 1) Sheet: 1 of 1

REFERENCE: 3.3.4/Q.774

TITLE: Syntactically invalid behavior; Invalid values for information elements

SUBTITLE: Begin message type; OTID length = 0

PURPOSE: To verify that on receipt of a corrupted Begin message, signalling point A is able to discard the message

PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. Arrange the data at SP B such that the Begin

message contains an OTID length of 0

CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP

EXPECTED MESSAGE SEQUENCE:

SP A (TSL) SP B (TSL)

:Detect syntax error <------ BEGIN

TEST DESCRIPTION

1. Arrange for SP B to send the corrupted Begin message to SP A, with an OTID length of 0.

2. CHECK A: THAT THE USER WAS NOT INFORMED OF THE BEGIN MESSAGE.

3. CHECK B: WERE NO MESSAGES SENT FROM SP A?

4. CHECK C: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE

STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets
Originating transaction ID tag: 01001000
Originating transaction ID length: 0
Originating transaction ID value: not present

Component portion tag: 01101100

TEST NUMBER: 1.2.1.1 2) Sheet: 1 of 1

REFERENCE: 3.3.4/Q.774

TITLE: Syntactically invalid behavior; Invalid values for information elements

SUBTITLE: Begin message type; OTID length > four octets

PURPOSE: To verify that signalling point A is able to deal with invalid encoding of OTID information element

PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. Arrange the data at SP B such that the Begin

message contains an OTID length of > four octets

CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP

EXPECTED MESSAGE SEQUENCE:

SP A (TSL) SP B (TSL)

:Detect syntax error BEGIN

TEST DESCRIPTION

- 1. Arrange for SP B to send the corrupted Begin message to SP A, with an OTID five octets long.
- 2. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A?
- 3. CHECK B: VERIFY THAT THE TR-USER AT SP A WAS NOT INFORMED OF THIS EVENT.
- 4. CHECK C: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE BEGIN MESSAGE.
- CHECK C: WERE ALL TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: five octests

Originating transaction ID value: OCTET STRING (5 octets long)

Component portion tag: 01101100 Component portion length: correct number of octets

TEST	EST NUMBER: 1.2.1.2 1)		Sheet: 1 of 2
REFERENCE: 3.3.4/Q.774			
TITLI	E: Syntactically invalid behavior; Invalid	values for information elements	
SUBT	TTLE: First Continue message DTID leng	gth = 0	
PURP	OSE: To verify that on receipt of a comessage or abort the transaction comessage.		length = 0, SP A is able to discard the
PRE-T		SP B (TSL) are to be in the idle state. Are contains a DTID of length = 0	range the data at SP B such that the first
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
	:Detect syntax error <		CONTINUE
	ABORT (P) (see Note)	>	
NOTE	E – If the Abort is not sent this may be valid	d behavior depending on the implementat	ion.
TEST DESCRIPTION			
1.	Arrange for SP A to send a Begin message to SP B.		
2.	Arrange for SP B to send the corrupted Continue message.		
3.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?		
4.	CHECK B: VERIFY THAT THE TR AT SP A.	-USER AT SP A WAS NOT INFORM	MED OF THE CONTINUE MESSAGE
5.		E MACHINES ASSOCIATED WITH T LEFT IN INITIATION SENT STATE ?	HIS TRANSACTION PRIOR TO THE
6.	CHECK D: IF AN ABORT MESSAG DTID AND P-ABORT CA		ECTLY FROM SP A WITH CORRECT

TEST NUMBER: 1.2.1.2 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets P-Abort cause value: incorrect transaction portion

TEST NUMBER: 1.2.1.3 1)			Sheet: 1 of 2
REFERENCE: 3.3.4/Q.774			
TITLI	E: Syntactically invalid behavior; I	nvalid values for information elements	
SUBT	ITLE: Subsequent Continue mess	age; Component portion length incorrect	
PURP		a corrupted Continue message with OTID derivent, SP A is able to abort the transaction	vable and DTID derivable and assigned,
PRE-T	TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state	
CONFIGURATION: 1		TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
SP A (TSL)			SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
		<	CONTINUE
	TR-CONTINUE Ind.		
	:Detect error	<	CONTINUE
	ABORT (P) (see Note)	>	
	TR-P-ABORT ind.		
NOTE	E – If the Abort is not sent this may	be valid behavior depending on the implementat	ion.
TEST	DESCRIPTION		
1.	Send a Begin message from SP A to SP B.		
2.	Arrange for SP B to send a correct Continue message to SP A.		
3.	Arrange for SP B to send a corrupted Continue message to SP A (incorrect CP length).		
4.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?		
5.	CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY RECEIVED AT SP A?		
6.	CHECK C: IF AN ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH CORRECT DTID AND P-ABORT CAUSE VALUE?		
7.	CHECK D: IF THE ABORT TRANSACTION L	WAS SENT, WERE TSL STATE MAC EFT IN THE IDLE STATE AT SP A?	CHINES ASSOCIATED WITH THIS

TEST NUMBER: 1.2.1.3 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE (1st)

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets

Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE (2nd)

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets

P-Abort cause value: badly formatted transaction portion 00000010

TEST NUMBER: 1.2.1.4 1) Sheet: 1 of 1 REFERENCE: 3.3.4/Q.774 TITLE: Syntactically invalid behavior; Invalid values for information elements SUBTITLE: End message; DTID length > four octets PURPOSE: To verify that on receipt of a corrupted End message, SP A is able to discard the message PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. Arrange the data at SP B such that the End message DTID length > four octets CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP EXPECTED MESSAGE SEQUENCE: SP A (TSL) SP B (TSL) TR-BEGIN req. **BEGIN** :Detect error TEST DESCRIPTION 1. Arrange for SP A to Send a Begin message to SP B. 2. Arrange for SP B to send a corrupted End message to SP A (invalid DTID length). CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? 3. 4. CHECK B: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE END MESSAGE? CHECK C: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION PRIOR TO THE END MESSAGE, LEFT IN THE INITIATION SENT STATE? CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: 00000101 (Invalid length) Destination transaction ID value: OCTET STRING (5 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

TEST	TEST NUMBER: 1.2.1.5 1)		Sheet: 1 of 2
REFERENCE: 3.3.4/Q.774			
TITLI	E: Syntactically invalid behavior;	Invalid values for information elements	
SUBT	TTLE: Abort message; Invalid P-	Abort cause value	
PURP	OSE: To verify that signalling p	point A is able to deal with incorrect encoding	g of P-Abort cause information element
PRE-	message w	and SP B (TSL) are to be in the idle state. Arrith a DTID that is derivable and assigned, contact the Begin message	ange the data at SP B such that an Abort ins a syntax error and is sent to SP A in
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
	:Detect syntax error	<	ABORT (P)
	TR-P-ABORT ind.		
NOTE – The sending of the TR-Abort ind. is implementation dependant.			
TEST DESCRIPTION			
1.	Arrange for SP A to send a Begin message to SP B and for SP B to respond with the corrupted Abort message. (Illegal P-Abort cause value).		
2.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?		
3.	CHECK B: VERIFY THAT N ABORT MESSAG	O MESSAGES ARE GENERATED BY SP A E.	IN RESPONSE TO THE CORRUPTED
4.		RT IND. WAS SENT, WERE TSL STATE M LEFT IN THE IDLE STATE AT SP A?	ACHINES ASSOCIATED WITH THIS

TEST NUMBER: 1.2.1.5 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)
(OTID value received in BEGIN message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets
P-Abort cause value: INTEGER (5 – Illegal value for this field)

TEST	NUMBER: 1.2.1.5 2)	Sheet: 1 of 2		
REFE	PRENCE: 3.3.4/Q.774			
TITLI	E: Syntactically invalid behavior; Invalid values for information elements			
SUBT	TITLE: Abort message; Invalid P-Abort cause length incorrect			
PURP	POSE: To verify that on receipt of a corrupted Abort message with incorrect cardiscard the message and advise the local user	ause length, signalling point A is able to		
PRE-	TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. At message with a DTID that is derivable and assigned, cont response to the Begin message	range the data at SP B such that an Abort ains a syntax error and is sent to SP A in		
	CONFIGURATION: 1 TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPE	ECTED MESSAGE SEQUENCE:			
	SP A (TSL)	SP B (TSL)		
	TR-BEGIN req>			
	BEGIN>			
	:Detect syntax error <	ABORT (P)		
	TR-P-ABORT ind.			
NOTI	E – The sending of the TR-Abort ind. is implementation dependant.			
TEST DESCRIPTION				
1.	Arrange for SP A to send a Begin message to SP B and for SP B to respond with P-Abort cause length).	the corrupted Abort message. (Corrupted		
2.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A	?		
3.	CHECK B: VERIFY THAT NO MESSAGES ARE GENERATED BY SP A ABORT MESSAGE.	IN RESPONSE TO THE CORRUPTED		
4.	CHECK C: IF THE TR-ABORT IND. WAS SENT, WERE TSL STATE M TRANSACTION LEFT IN THE IDLE STATE AT SP A?	IACHINES ASSOCIATED WITH THIS		

TEST NUMBER: 1.2.1.5 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID tag. Vivorovi

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets (i.e. not one) P-Abort cause value: INTEGER (0 .. 4)

TEST 1	NUMBER: 1.2.2.1 1)		Sheet: 1 of 1
REFER	RENCE: 3.3.4/Q.774		
TITLE	Syntactically invalid behavior;	Invalid structure	
SUBTI	TLE: Unidirectional message Ty	rpe; Unknown information element present	
PURPO	OSE: To verify that on receipt of	a corrupted Unidirectional message, signalling p	oint A is able to discard the message
PRE-T	EST CONDITIONS: SP A (TS) Unidirection	L) and SP B (TSL) are to be in the idle state.	Arrange the data at SP B such that an to SP A
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPEC	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	:Detect syntax error	<	UNIDRECTIONAL
TEST I	DESCRIPTION		
1.	Arrange for SP B to send the corr	rupted Unidirectional message to SP A.	
2.	CHECK A: VERIFY THAT TO SP A.	HE TR-USER WAS NOT INFORMED OF TH	IE UNIDIRECTIONAL MESSAGE AT
3.	CHECK B: VERIFY THAT N MESSAGE.	O MESSAGES WERE GENERATED IN RES	SPONSE TO THE UNIDIRECTIONAL
4.	CHECK C: WERE TSL STAT STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE
CHECI	K TABLE FOR INFORMATION	ELEMENTS WITHIN MESSAGES	
UNIDE	RECTIONAL		
Me Me	ssage type tag: 01100001 ssage type length: correct number	er of octets	
Component portion missing			

TEST NUMBER: 1.2.2.2 1)			Sheet: 1 of 1		
REFE	ERENCE: 3.3.4/Q.774				
TITL	E: Syntactically invalid behavior;	Invalid structure			
SUBT	FITLE: Begin message type; OTII) absent			
PURF	POSE: To verify that on receipt of	a corrupted Begin message; signalling point A is	s able to discard the message		
PRE-	TEST CONDITIONS: SP A (TSI message co	L) and SP B (TSL) are to be in the idle state. Arrontains a syntax error and the OTID is not deriva	ange the data at SP B such that the Begin ble		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPE	ECTED MESSAGE SEQUENCE:				
	SP A (TSL)		SP B (TSL)		
	:Detect syntax error	<	BEGIN		
TEST	DESCRIPTION				
1.	Arrange for SP B to send the cor	rupted Begin message to SP A, with OTID not pr	resent.		
2.	CHECK A: VERIFY THAT TI	HE TR-USER WAS NOT INFORMED OF THIS	S EVENT AT SP A.		
3.	CHECK B: VERIFY THAT N BEGIN MESSAGE	NO MESSAGES WERE GENERATED IN RE E.	SPONSE TO THE THE CORRUPTED		
4.	CHECK C: WERE TSL STATE STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE		
СНЕС	CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES				
BEGI	N				
M M	lessage type tag: 01100010 lessage type length: correct number	er of octets			
O	TID absent				

TEST	NUMBER: 1.2	2.2.2 2)		Sheet: 1 of 2
REFE	RENCE: 3.3.4	/Q.774		
TITLI	E: Syntactically	Invalid Behavior;	Invalid structure	
SUBT	TTLE: Begin N	Message Type; Unki	nown information element present	
PURP	OSE: To verify able to di	y that on receipt of iscard the message a	a corrupted Begin message, with an invalid in and generate an Abort message	formation element, signalling point A is
PRE-T	TEST CONDITI		and SP B (TSL) are to be in the idle state. Arrantains a syntax error	inge the data at SP B such that the Begin
	CONFIGURA	TION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSA	GE SEQUENCE:		
	SP A (T	SL)		SP B (TSL)
	:Detect synte	ax error	<	BEGIN
	ABORT (P	')	>	
NOTE	E – If the Abort i	is not sent, this may	be valid behavior depending on the implementa	tion.
TEST	DESCRIPTION	N		
1.	Arrange for SI	P B to send the corr	upted Begin message to SP A, with an invalid in	formation element after the OTID.
2.	CHECK A: IF AN ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH CORRECT DTID AND CORRECT P-ABORT CAUSE VALUE?			
3.		WERE TSL STAT STATE AT SP A?	TE MACHINES ASSOCIATED WITH THIS	TRANSACTION LEFT IN THE IDLE

TEST NUMBER: 1.2.2.2 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Information element tag: unknown (eg. 01101101) Information element length: correct number of octets Information element value: OCTET STRING

ABORT (P)

Message type tag: 01100111 Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets

P-Abort cause value: incorrect transaction portion 00000011

TEST NUMBER: 1.2.2.3 1)			Sheet: 1 of 2
REFE	RENCE: 3.3.4/Q.774		
TITLI	E: Syntactically Invalid Behavio	or; Invalid structure	
SUBT	TITLE: First Continue Message	OTID absent	
PURP	OSE: To verify that on receipt	of a corrupted Continue message, signalling point A	A is able to discard the message
PRE-	receipt o	SL) and SP B (TSL) are to be in the idle state. SP Is the Begin message. Arrange the data at SP B surprise and the OTID is not derivable	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE	:	
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
	:Detect syntax error	<	CONTINUE
TEST	DESCRIPTION		
1.	Arrange for SP A to send a Be	gin message to SP B.	
2.	Arrange for SP B to send the corrupted Continue message (OTID not derivable) to SP A.		
3.	CHECK A: WAS THE BEG	IN MESSAGE CORRECTLY SENT FROM SP A	?
4.	CHECK B: VERIFY THAT	THE TR-USER WAS NOT INFORMED OF THE	CONTINUE MESSAGE AT SP A?
5.	CHECK C: VERIFY THAT CONTINUE MI	NO MESSAGES WERE GENERATED BY SP A ESSAGE?	IN RESPONSE TO THE CORRUPTED
6.		CATE MACHINES ASSOCIATED WITH THE ESSAGE, LEFT IN THE INITIATION SENT STA	

TEST NUMBER: 1.2.2.3 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

OTID absent

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)
(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

TEST	NUMBER: 1.2.2.3 2)		Sheet: 1 of 2
REFE	RENCE: 3.3.4/Q.774		
TITLI	E: Syntactically Invalid Behavior;	Invalid structure	
SUBT	TTLE: First Continue Message; D	TID absent	
PURP	OSE: To verify that on receipt of the message or abort the tran	a corrupted Continue message containing no Disaction	ΓΙD, signalling point A is able to discard
PRE-T	TEST CONDITIONS: SP A (TSL receipt of the DTID	and SP B (TSL) are to be in the idle state. SP B ee Begin message. Arrange the data at SP B suc	3 to respond with a Continue message on h that the Continue message contains no
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
	:Detect syntax error	<	CONTINUE
		(DTID absent)	
	ABORT (P)	>	
NOTE	E – If the Abort is not sent, this may	be valid behavior depending on the implementa	tion.
TEST	DESCRIPTION		
1.	Arrange for SP A to send a Begin	message to SP B.	
2.	Arrange for SP B to send the corrupted Continue message (DTID absent).		
3.	CHECK A: WAS THE BEGIN	MESSAGE CORRECTLY SENT FROM SP A	?
4.	CHECK B: VERIFY THAT TH	IE TR-USER WAS NOT INFORMED OF THE	CONTINUE MESSAGE AT SP A?
5.		STATE MACHINES ASSOCIATED WITH T AGE, LEFT IN THE INITIATION SENT STA	
6.	CHECK D: IF AN ABORT M DTID AND CORR	ESSAGE WAS SENT, WAS IT SENT CORRE ECT P-ABORT CAUSE VALUE?	ECTLY FROM SP A WITH CORRECT

TEST NUMBER: 1.2.2.3 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

DTID absent

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets

P-Abort cause value: incorrect transaction portion 00000011

TEST	NUMBER: 1.2.2.3 3)		Sheet: 1 of 2
REFE	RENCE: 3.3.4/Q.774		
TITLI	E: Syntactically Invalid Behavior;	Invalid structure	
SUBT	TITLE: First Continue Message; C	TID duplicated	
PURP	OSE: To check the correct behat duplicated OTID	vior of the implementation under test on rece	ipt of a first Continue message with a
PRE-	TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CCTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
		(with duplicated OTID)	
		<	CONTINUE
	ABORT (P)	>	
	TR-P-ABORT ind. <======		
NOTE	E – If the ABORT message and prin	mitive are not sent, this may be valid behavior de	pending on the implementation.
TEST	DESCRIPTION		
1.	Arrange SP A to send a Begin me	essage.	
2.	Arrange for SP B to send a Conti	nue message to SP A with a duplicated OTID.	
3.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?		
4.	CHECK B: IF AN ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH CORRECT DTID VALUE AND CORRECT P-ABORT CAUSE VALUE?		
5.	CHECK C: IF THE ABORT ASSOCIATED WI	MESSAGE AND PRIMITIVE WERE SEN TH THIS TRANSACTION LEFT IN THE IDLI	IT, WERE TSL STATE MACHINES E STATE AT SP A?

TEST NUMBER: 1.2.2.3 3) Sheet: 2 of 2 CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long) Component portion tag: 01101100 Component portion length: correct number of octets CONTINUE Message type tag: 01100101 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long) Duplicated Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) Destination transaction ID tag: 01001001 Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message) Component portion tag: 01101100 Component portion length: correct number of octets ABORT (P) Message type tag: 01100111 Message type length: correct number of octets Destination transaction ID tag: 01001001 Destination transaction ID length: correct number of octets Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in CONTINUE message) P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets

P-Abort cause value: incorrect transaction portion 00000011

TEST	NUMBER: 1.2.2.3 4)		Sheet: 1 of 2
REFE	CRENCE: 3.3.4/Q.774		
TITL	E: Syntactically Invalid Behavior;	Invalid structure	
SUBT	TITLE: First Continue Message; D	TID duplicated	
PURF	POSE: To check the correct behat duplicated DTID	vior of the implementation under test on rece	cipt of a first Continue message with a
PRE-	TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CCTED MESSAGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
	TR-BEGIN req.		
	BEGIN	>	
		(with duplicated DTID)	CONTINUE
	ABORT (P)	>	CONTINUE
	TR-P-ABORT ind.		
	<========		
NOTI	E – If the ABORT message and prin	mitive are not sent, this may be valid behavior de	pending on the implementation.
TEST	DESCRIPTION		
1.	Arrange SP A to send a Begin me	essage.	
2.	Arrange for SP B to send a Continue message to SP A with a duplicated DTID.		
3.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?		
4.	CHECK B: WAS AN ABORT MESSAGE WITH CORRECT DTID VALUE AND CORRECT P-ABORT CAUSE VALUE CORRECTLY SENT FROM SP A?		
5.	CHECK C: IF THE ABORT ASSOCIATED WI	MESSAGE AND PRIMITIVE WERE SEN TH THIS TRANSACTION LEFT IN THE IDLI	IT, WERE TSL STATE MACHINES E STATE AT SP A?

TEST NUMBER: 1.2.2.3 4) Sheet: 2 of 2 CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long) Component portion tag: 01101100 Component portion length: correct number of octets CONTINUE Message type tag: 01100101 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long) Destination transaction ID tag: 01001001 Destination transaction ID length: correct number of octets Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message) Duplicated Destination transaction ID tag: 01001001 Destination transaction ID length: correct number of octets Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message) Component portion tag: 01101100 Component portion length: correct number of octets ABORT (P) Message type tag: 01100111 Message type length: correct number of octets Destination transaction ID tag: 01001001 Destination transaction ID length: correct number of octets Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in CONTINUE message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets

P-Abort cause value: incorrect transaction portion 00000011

TEST	NUMBER: 1.2.2.3 5)		Sheet: 1 of 2		
REFE	RENCE: 3.3.4/Q.774				
TITL	E: Syntactically Invalid Behavior;	Invalid structure			
SUBT	TITLE: First Continue Message; U	Inknown information element present			
PURF	POSE: To verify that on receipt of	a corrupted Continue message, signalling point A	A behaves correctly		
PRE-	message wi) and SP B (TSL) are to be in the idle state. Arra th an OTID that is derivable and a DTID that is sent to SP A in response to the Begin message	nge the data at SP B such that a Continue derivable and assigned, contains a syntax		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPE	CTED MESSAGE SEQUENCE:				
	SP A (TSL)		SP B (TSL)		
	TR-BEGIN req.				
	BEGIN	>			
	:Detect syntax error	<	CONTINUE		
	ABORT (P)	>			
	TR-P-ABORT ind.				
NOTI	E – If the ABORT message and prin	mitive are not sent, this may be valid behavior de	pending on the implementation.		
TEST	DESCRIPTION				
1.	Arrange for SP A to send a Begin	n message to SP B.			
2.	Arrange for SP B to send the co element (eg P-Abort Cause).	rrupted Continue message with an extra informa	ation element after the DTID information		
3.	CHECK A: VERIFY THAT TI	HE TR-USER WAS NOT INFORMED OF THE	CONTINUE MESSAGE AT SP A?		
4.	DTID AND THE (MESSAGE WAS SENT, WAS IT SENT CORRI CORRECT P-ABORT CAUSE VALUE? ANSACTION PORTION)	ECTLY FROM SP A, WITH CORRECT		
5.	CHECK C: IF THE MESSA	GE AND PRIMITIVE ABORT WERE SEN TH THIS TRANSACTION LEFT IN THE IDLI			

TEST NUMBER: 1.2.2.3 5) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Information element tag: unknown (eg. 01101101) Information element length: correct number of octets Information element value: OCTET STRING

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets
P-Abort cause value: incorrect transaction portion 00000011

TEST NUMBER: 1.2.2.4 1)		Sheet: 1 of 2		
REFE	ERENCE: 3.3.4	-/Q.774		
TITL	E: Syntactically	y Invalid Behavior; I	nvalid structure	
SUBT	FITLE: Subsequ	uent Continue Messa	age; OTID absent	
PURF	POSE: To verify message		corrupted Continue message after transaction	establishment, SP A is able to discard the
PRE-	TEST CONDITI	IONS: SP A (TSL)	and SP B (TSL) are to be in the idle state	
	CONFIGURA	ATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	ECTED MESSA	GE SEQUENCE:		
	SP A (T	SL)		SP B (TSL)
	TR-BEGIN i	req.		
		===>		
	BEGIN		>	
			<	CONTINUE
	TR-CONTIN	VUE ind.		
	:Detect erro	an.	<	CONTINUE
.2666.676.			COMMINGE	
TEST	DESCRIPTION	N		
1.	Send a Begin	message from SP A	to SP B.	
2.	Arrange for SP B to send a correct Continue message to SP A.			
3.	. Arrange for SP B to send a corrupted Continue message to SP A (OTID not derivable).			
4.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?			
5.	CHECK B: WAS THE FIRST CONTINUE MESSAGE CORRECTLY RECEIVED AT SP A?			
6.	CHECK C: VERIFY THAT THE TR-USER AT SP A WAS NOT INFORMED OF THE CORRUPTED CONTINUE MESSAGE?			ED OF THE CORRUPTED CONTINUE
7.	CHECK D: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE CORRUPTED CONTINUE MESSAGE?			IN RESPONSE TO THE CORRUPTED
8.			TE MACHINES ASSOCIATED WITH THI TINUE MESSAGE, LEFT IN THE ACTIVE S	

TEST NUMBER: 1.2.2.4 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE (1st)

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE (2nd)

Message type tag: 01100101

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

TEST NUMBER: 1.2.2.4 2)			Sheet: 1 of 2		
REFE	PRENCE: 3.3.4/Q.774				
TITL	E: Syntactically Invalid Behavior; I	nvalid structure			
SUBT	TITLE: Subsequent Continue Messa	age; Unknown information element present			
PURF	POSE: To verify that on receipt of after transaction establishmen	a corrupted Continue message with OTID derint, SP A behaves correctly	vable and DTID derivable and assigned,		
PRE-	TEST CONDITIONS: SP A (TSL)	and SP B (TSL) are to be in the idle state			
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPE	CCTED MESSAGE SEQUENCE:				
	SP A (TSL)		SP B (TSL)		
		<	BEGIN		
	TR-BEGIN ind. <======				
	TR-CONTINUE req.				
		>	CONTINUE		
	:Detect error	<	CONTINUE		
	ABORT (P)	>			
	TR-P-ABORT Ind.				
	<				
NOTI	E – If the ABORT message and prim	itive are not sent, this may be valid behavior de	pending on the implementation.		
TEST	DESCRIPTION				
1.	Send a Begin message from SP B	to SP A.			
2.	Arrange for SP A to send a correct Continue message to SP B.				
3.	Arrange for SP B to send a corrupted Continue message to SP A (extra Information Element after the DTID Information Element).				
4.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A?				
5.	CHECK B: WAS THE FIRST C	CONTINUE MESSAGE CORRECTLY SENT I	FROM SP A?		
6.		SSAGE WAS SENT, WAS IT SENT CORR ECT P-ABORT CAUSE VALUE?	ECTLY FROM SP A WITH CORRECT		
7.		WAS SENT, WERE TSL STATE MAG EFT IN IDLE STATE AT SP A?	CHINES ASSOCIATED WITH THIS		

TEST NUMBER: 1.2.2.4 2) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE (1st)

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets

Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUED (2nd)

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

(OTID value used in BEGIN message)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

Information element tag: unknown (eg. 01101101) Information element length: correct number of octets
Information element value: OCTET STRING

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets

P-Abort cause value: incorrect transaction portion 00000011

TEST NUMBER: 1.2.2.5 1) Sheet: 1 of 1 REFERENCE: 3.3.4/Q.774 TITLE: Syntactically Invalid Behavior; Invalid structure SUBTITLE: End Message; DTID absent PURPOSE: To verify that on receipt of a corrupted End message, SP A is able to discard the message PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. Arrange the data at SP B such that the End message contains a syntax error (DTID absent) CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP EXPECTED MESSAGE SEQUENCE: SP A (TSL) SP B (TSL) TR-BEGIN req. **BEGIN** :Detect syntax error END TEST DESCRIPTION 1. Arrange for SP A to send a Begin message to SP B. 2. Arrange for SP B to send a corrupted End message to SP A.(DTID absent.) 3. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? 4. CHECK B: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE MESSAGE AT SP A? 5. CHECK C: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE CORRUPTED END MESSAGE? CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION, PRIOR TO THE END MESSAGE, LEFT IN THE INITIATION SENT STATE AT SP A? CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) Component portion tag: 01101100 Component portion length: correct number of octets Message type tag: 01100100 Message type length: correct number of octets DTID absent Component portion tag: 01101100 Component portion length: correct number of octets

TEST NUMBER: 1.2.2.6 1) Sheet: 1 of 1 REFERENCE: 3.3.4/Q.774 TITLE: Syntactically Invalid Behavior; Invalid structure SUBTITLE: Abort Message; DTID absent PURPOSE: To verify that on receipt of a corrupted Abort message, SP A is able to discard the message PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. Arrange the data at SP B such that the Abort message contains a syntax error (DTID absent) CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP EXPECTED MESSAGE SEQUENCE: SP A (TSL) SP B (TSL) TR-BEGIN req. BEGIN **ABORT** :Detect syntax error TEST DESCRIPTION Arrange for SP A to send a Begin message to SP B. 1. 2. Arrange for SPB to send a corrupted Abort message to SPA. 3. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? CHECK B: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE MESSAGE AT SP A? 4. 5. CHECK C: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE CORRUPTED ABORT MESSAGE? CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION, PRIOR TO THE ABORT MESSAGE, LEFT IN THE INITIATION SENT STATE AT SP A? CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long) Component portion tag: 01101100 Component portion length: correct number of octets ABORT (P) Message type tag: 01100111 Message type length: correct number of octets DTID absent P-Abort cause tag: 01101100 P-Abort cause length: correct number of octets P-Abort cause value: eg. incorrect transaction portion 00000011

TEST NUMBER: 1.2.2.7 1) Sheet: 1 of 1

REFERENCE: 3.3.4/Q.774

TITLE: Syntactically Invalid Behavior; Invalid structure

SUBTITLE: Unknown Message; OTID not included

PURPOSE: To verify that on receipt of an Unknown message, signalling point A is able to discard the message

PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. Arrange the data at SP B such that an

Unknown message with an OTID that is not derivable is sent to SP A

CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP

EXPECTED MESSAGE SEQUENCE:

message type

SP A (TSL) SP B (TSL)

:Detect Unknown <------ UNKNOWN MESSAGE

TEST DESCRIPTION

- 1. Arrange for SP B to send the Unknown message to SP A.
- 2. CHECK A: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THIS EVENT AT SP A?
- 3. CHECK B: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE UNKNOWN MESSAGE?
- 4. CHECK C: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

UNKNOWN MESSAGE

Message type tag: unknown (eg 01100110) Message type length: correct number of octets

OTID absent

TEST	NUMBER: 1.2.2.7 2)		Sheet: 1 of 1	
REFE	ERENCE: 3.3.4/Q.774			
TITL	E: Syntactically Invalid Behavio	r; Invalid structure		
SUBT	FITLE: Unknown Message; OTI	D included and DTID not included		
PURI	POSE: To verify that on receipt of	of an Unknown message, signalling point A behave	es correctly	
PRE-		L) and SP B (TSL) are to be in the idle state. Are with an OTID that is derivable and a DTID that is SP A		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPI	ECTED MESSAGE SEQUENCE			
	SP A (TSL)		SP B (TSL)	
	:Detect Unknown. message type	<	UNKNOWN MESSAGE	
	ABORT (P)	>		
NOT	E – If the Abort message is not se	nt, this may be valid behavior depending on the im	plementation.	
TEST	DESCRIPTION			
1.	Arrange for SP B to send the U	nknown message to SP A.		
2.		MESSAGE WAS SENT, WAS IT SENT CO AND CORRECT P-ABORT CAUSE VALUE?	DRRECTLY FROM SP A WITH THE	
3.	3. CHECK B: IF THE ABORT WAS SENT, WERE TSL THE STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?			
CHE	CK TABLE FOR INFORMATIO	N ELEMENTS WITHIN MESSAGES		
UNKNOWN MESSAGE Message type tag: unknown (eg 01100110) Message type length: correct number of octets				
O	Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)			
	RT (P)			
M	Message type tag: 01100111 Message type length: correct number of octets			
D	estination transaction ID tag: 01 estination transaction ID length: estination transaction ID value:			
P	-Abort cause tag: 01001010 -Abort cause length: one octet -Abort cause value: unrecognize	d message type 00000000		

TEST NUMBER: 1.2.2.7 3)			Sheet: 1 of 2	
REFERENCE: 3.3.4/Q.774				
TITLI	E: Syntactically Invalid Behavior;	Invalid structure		
SUBT	TTLE: Unknown Message; OTID	included and DTID included		
PURP	OSE: To verify that on receipt of	an Unknown message with assigned DTID, SP A	A is able to behave correctly	
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. Arrange the data at SP B such than an Unknown message with an OTID that is derivable and a DTID that is derivable and assigned is sent to SP A in response to the Begin message				
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE SEQUENCE:			
	SP A (TSL)		SP B (TSL)	
	TR-BEGIN req.			
	BEGIN	>		
	:Detect Unknown message type	<	UNKNOWN MESSAGE	
	ABORT (P)			
	TR-P-ABORT ind. <======	>		
NOTE – If the ABORT message and primitive are not sent, this may be valid behavior depending on the implementation.				
TEST DESCRIPTION				
1.	Arrange for SP A to send a Begin message to SP B and for SP B to respond with the Unknown message.			
2.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?			
3.	CHECK B: IF THE P-ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH THE CORRECT DTID AND CORRECT P-ABORT CAUSE VALUE?			
4.	CHECK C: IF THE ABORT WAS SENT, WAS THE TR-USER AT SP A ADVISED BY A TR-P-ABORT INDICATION PRIMITIVE THAT THIS TRANSACTION HAD BEEN ABORTED?			
5.		WAS SENT, WERE TSL STATE MAC LEFT IN THE IDLE STATE AT SP A?	CHINES ASSOCIATED WITH THIS	

TEST NUMBER: 1.2.2.7 3) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

UNKNOWN MESSAGE

Message type tag: unknown (eg 01100110) Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in UNKNOWN message)

P-Abort cause tag: 01001010 P-Abort cause length: one octet

P-Abort cause value: unrecognized message type 00000000

TEST NUMBER: 1.2.3.1 1) Sheet: 1 of 1 REFERENCE: 3.3.4/Q.774 TITLE: Syntactically Invalid Behavior; Invalid enconding SUBTITLE: Begin Message Type; Invalid tag PURPOSE: To verify that on receipt of a corrupted Begin message with Invalid tag, signalling point A behaves correctly PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. Arrange the data at SP B that the Begin message contains an Invalid tag TYPE OF TEST: VAT TYPE OF SP: SP CONFIGURATION: 1 EXPECTED MESSAGE SEQUENCE: SP A (TSL) SP B (TSL) :Detect syntax error **BEGIN** ABORT (P) NOTE – If the Abort message is not sent, this may be valid behavior depending on the implementation. TEST DESCRIPTION Arrange for SP B to send the corrupted Begin message to SP A. 1. CHECK A: CHECK THAT THE USER WAS NOT INFORMED OF THE BEGIN MESSAGE? 2. CHECK B: WERE THE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE 3. STATE AT SP A? 4.. CHECK C: IF AN ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH CORRECT DTID AND CORRECT P-ABORT CAUSE VALUE? CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES **BEGIN** Message type tag: 01100010 Message type length: correct number of octets Invalid tag: eg. 00100010 Originating transaction ID tag: 01001000 Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) ABORT (P) Message type tag: 01100111 Message type length: correct number of octets Destination transaction ID tag: 01001001 Destination transaction ID length: correct number of octets Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message) P-Abort cause tag: 01001010 P-Abort cause length: correct number of octets P-Abort cause value: incorrect transaction portion 00000011

TEST	NUMBER: 1.2.3.2 1)	Sheet: 1 of 2		
REFE	RENCE: 3.3.4/Q.774			
TITLI	E: Syntactically Invalid Behavior;	Invalid encoding		
SUBT	TTLE: Continue Message; Invalid	tag		
PURP	OSE: To verify that on receipt of	a corrupted Continue message with Invalid tag, s	signalling point A behaves correctly	
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state. SP B to respond with a Continue message on receipt of the Begin message. Arrange the data at SP B such that Continue message contains a syntax error (invalid tag)				
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE SEQUENCE:			
	SP A (TSL)		SP B (TSL)	
	TR-BEGIN req. =====>			
	BEGIN	>		
	:Detect syntax error < CONTINUE			
	ABORT (P)			
	TR-P-ABORT ind.			
NOTE – If the ABORT message and the primitive are not sent, this may be valid behavior depending on the implementation.				
TEST DESCRIPTION				
1.	Arrange for SP A to send a Begin message to SP B.			
2.	Arrange for SP B to send the corrupted Continue message to SP A.			
3.	CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A?			
4.	CHECK B: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE CONTINUE MESSAGE AT SP A?			
5.		ESSAGE WAS SENT, WAS IT SENT CORRI ECT P-ABORT CAUSE VALUE?	ECTLY FROM SP A WITH CORRECT	

TEST NUMBER: 1.2.3.2 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets Originating transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Invalid tag: eg. 00011111

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets

Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in BEGIN message)

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

P-Abort cause tag: 01001010

P-Abort cause length: correct number of octets

P-Abort cause value: incorrect transaction portion 00000011

TEST NUMBER: 1.3.1 1)				Sheet: 1 of 2
REFERENCE: 3.3.4/Q.774				
TITLI	E: Incorporate	e Messages; Continue	e Message Type	
SUBT	TITLE: Receip	pt of Continue messa	ge in idle state with unassigned DTID	
PURP	POSE: To veri messag	ify that on receipt of e and generate an Ab	f a Continue message with unassigned DTID, aport message	signalling point A is able to discard the
PRE-TEST CONDITIONS: SP A (TSL) to be in the idle state and SP B (TSL) to be in the IR/Active state. Arrange the data at SP B such that a Continue message with an OTID that is derivable and a DTID that is derivable but unassigned is sent to SP A				
	CONFIGUR	ATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CCTED MESS	AGE SEQUENCE:		
	SP A (TSL)		SP B (TSL)
			<	CONTINUE
	ABORT ((P)	>	
TEST	DESCRIPTIO	ON		
1.	Arrange for SP B to send the Continue message with unassigned DTID to SP A.			
2.	CHECK A: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE CONTINUE MESSAGE AT SP A?			
3.	CHECK B: WAS THE DTID IN THE ABORT MESSAGE EQUAL TO THE OTID IN THE CONTINUE MESSAGE?			
4.	CHECK C: WAS AN ABORT MESSAGE CORRECTLY SENT FROM SP A WITH A P-ABORT CAUSE VALUE OF UNRECOGNIZED TRANSACTION ID?			
5.	CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A?			

TEST NUMBER: 1.3.1 1) Sheet: 2 of 2

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000
Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long)

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

(OTID value received in CONTINUE message)

P-Abort cause tag: 01001010

P-Abort cause length: one octet P-Abort cause value: unrecognized transaction ID 00000001

TEST NUMBER: 1.3.2 1) Sheet: 1 of 1

REFERENCE: 3.3.4/Q.774

TITLE: Incorporate Messages; End Message Type

SUBTITLE: Receipt of End message in idle state

PURPOSE: To verify that on receipt of an End message with unassigned DTID, signalling point A is able to discard the message

PRE-TEST CONDITIONS: SP A (TSL) to be in the idle state and SP B (TSL) to be in the IR/Active state. Arrange the data at SP

B such that an End message with a DTID that is derivable but unassigned is sent to SP A

CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP

EXPECTED MESSAGE SEQUENCE:

SP A (TSL) SP B (TSL)

<----- END

TEST DESCRIPTION

1. Arrange for SP B to send the End message with unassigned DTID to SP A.

2. CHECK A: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE END MESSAGE AT SP A?

3. CHECK B: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE END MESSAGE?

4. CHECK C: WERE TSL STATE MACHINES ASSOCIATED WITH THE TRANSACTION LEFT IN THE IDLE STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long)

Component portion tag: 01101100

Component portion length: correct number of octets

TEST NUMBER: 1.3.3 1) Sheet: 1 of 1

REFERENCE: 3.3.4/Q.774

TITLE: Incorporate Messages; Abort Message Type

SUBTITLE: Receipt of Abort message in idle state

PURPOSE: To verify that on receipt of an Abort message with unassigned DTID, signalling point A is able to discard the

message

PRE-TEST CONDITIONS: SP A (TSL) to be in the idle state and SP B (TSL) to be in the IR/Active state. Arrange the data at SP

B such that an Abort message with a DTID that is derivable but unassigned is sent to SP A

TYPE OF TEST: VAT TYPE OF SP: SP CONFIGURATION: 1

EXPECTED MESSAGE SEQUENCE:

SP A (TSL) SP B (TSL)

ABORT (P)

TEST DESCRIPTION

1. Arrange for SP B to send the Abort message with unassigned DTID to SP A.

CHECK A: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE ABORT MESSAGE AT SP A? 2.

CHECK B: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE ABORT 3.

MESSAGE?

CHECK C: WERE ALL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE

STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)

P-Abort cause tag: 01001010

P-Abort cause length: one octet P-Abort cause value: INTEGER {0, 1, 2, 3, 4}

TEST NUMBER: 1.4.1 1)	Sheet: 1 of 3			
REFERENCE: 3.3.3.2/Q.774				
TITLE: Multiple Transaction Encoding;	TITLE: Multiple Transaction Encoding; Valid Transaction Encoding			
SUBTITLE: New transaction request du	SUBTITLE: New transaction request during transaction establishment			
PURPOSE: To verify that the signalling transaction	point A is able to correctly react to a Begin me	ssage during the establishment of another		
PRE-TEST CONDITIONS: SP A (TSL)	and SP B (TSL) are to be in the idle state			
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE SEQUENCE:				
SP A (TSL)		SP B (TSL)		
TR-BEGIN req. =====>				
BEGIN	>			
TR-BEGIN ind. <======	<	BEGIN (new transaction)		
TR-END req. ====>				
(Basic) (end new transaction)				
END	>			
TR-END ind. <=======	<	END		

			_
TEST	ST NUMBER: 1.4.1 1)		Sheet: 2 of 3
TEST	TEST DESCRIPTION		
1.	Arrange for	SP A to send a Begin message to SP B.	
2.	Arrange for SP B to send a Begin message to SP A (new transaction).		
3.	Arrange for SP A to respond with an End message to the 2nd Begin message.		
4.	Arrange for SP B to respond with an End message to the 1st Begin message.		
5.	CHECK A: WAS THE FIRST BEGIN MESSAGE CORRECTLY SENT BY SP A?		
6.	CHECK B: WAS THE SECOND BEGIN MESSAGE CORRECTLY RECEIVED BY SP A?		
7.	CHECK C: WAS THE DTID IN THE FIRST END MESSAGE THE SAME AS THE OTID IN THE SECOND BEGIN MESSAGE?		
8.	CHECK D:	WAS THE SECOND END MESSAGE CORRECTLY RECEIVED	DBY SP A?
9.	CHECK E:	WERE TSL STATE MACHINES ASSOCIATED WITH THESE STATE AT SP A?	TRANSACTIONS LEFT IN THE IDLE

104

TEST NUMBER: 1.4.1 1) Sheet: 3 of 3

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN (1st)

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets

Originating transaction ID value: OCTET STRING (1-4 octets long) X

Component portion tag: 01101100

Component portion length: correct number of octets

BEGIN (2nd)

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) Y

Component portion tag: 01101100

Component portion length: correct number of octets

END (1st)

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long) Y

(OTID value received in 2nd BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

END (2nd)

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) X

(OTID value received in 1st BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

TEST NUMBER: 1.4.1 2)		Sheet: 1 of 3			
REFERENCE: 3.3.3.2/Q.774	REFERENCE: 3.3.3.2/Q.774				
TITLE: Multiple Transaction Encoding;	Valid Transaction Encoding				
SUBTITLE: New transaction request aft	ter transaction establishment				
PURPOSE: To verify that the signalling transaction	point A is able to correctly react to a Begin m	essage after the establishment of another			
PRE-TEST CONDITIONS: SP A (TSL)) and SP B (TSL) are to be in the idle state				
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP			
EXPECTED MESSAGE SEQUENCE:					
SP A (TSL)		SP B (TSL)			
TR-BEGIN req.					
BEGIN	>				
	<	CONTINUE			
TR-CONTINUE ind.					
	<	BEGIN (new transaction)			
TR-BEGIN ind. <=======					
TR-END req.					
(Basic) (end new transaction)					
END	>				
	<	END			
TR-END ind. <======					

TEST NUMBER: 1.4.1 2) Sheet: 2 of 3

TEST DESCRIPTION

- 1. Arrange for SP A to send a Begin message to SP B.
- 2. Arrange for SP B to respond with a Continue message to Begin message.
- 3. Arrange for SP B to send a Begin message to SP A (new transaction).
- 4. Arrange for SP A to respond with an End message to the 2nd Begin message.
- 5. Arrange for SP B to respond with an End message to the 1st Begin message.
- CHECK A: WAS THE FIRST BEGIN MESSAGE CORRECTLY SENT BY SP A? 6.
- 7. CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY RECEIVED BY SP A?
- CHECK C: WAS THE SECOND BEGIN MESSAGE CORRECTLY RECEIVED BY SP A? 8.
- 9. CHECK D: WAS THE DTID IN THE FIRST END MESSAGE THE SAME AS THE OTID IN THE SECOND BEGIN MESSAGE?
- 10. CHECK E: WAS THE SECOND END MESSAGE CORRECTLY RECEIVED BY SP A?
- 11. CHECK F: WERE TSL STATE MACHINES ASSOCIATED WITH THESE TRANSACTIONS LEFT IN THE IDLE STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN (1st)

Message type tag: 01100010

Message type length: correct number of octets Destination transaction ID tag: 01001000

Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long) X

Component portion tag: 01101100

Component portion length: correct number of octets

TEST NUMBER: 1.4.1 2) Sheet: 3 of 3

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets

Originating transaction ID value: OCTET STRING (1-4 octets long) Y

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) X

(OTID value received in 1st BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

BEGIN (2nd)

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) Z

Component portion tag: 01101100

Component portion length: correct number of octets

END (1st)

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) Z

(OTID value received in 2nd BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

END (2nd)

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) X

(OTID value received in 1st BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

TEST NUMBER: 1.4.2 1)	Sheet: 1 of 3	
REFERENCE: 3.3.3.2/Q.774		
TITLE: Multiple Transaction Encoding;	Inopportune Messages	
SUBTITLE: Message with unassigned I	OTID during transaction establishment	
PURPOSE: To verify that the signalling the establishment of another	g point A is able to correctly react to a Continu- transaction	e message with DTID unassigned during
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPECTED MESSAGE SEQUENCE:		
SP A (TSL)		SP B (TSL)
TR-BEGIN req. =====>		
BEGIN	>	
	<	CONTINUE (new transaction)
ABORT (P)	>	
TR-P-ABORT ind. <======		
TR-END ind.	<	END
<==========		

TEST NUMBER: 1.4.2 1) Sheet: 2 of 3 TEST DESCRIPTION 1. Arrange for SP A to send a Begin message to SP B. 2. Arrange for SP B to send a Continue message with unassigned DTID to SP A. 3. Arrange for SP B to respond with an End message to the Begin message. 4. CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT BY SP A? 5. CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY RECEIVED BY SP A? 6. CHECK C: WAS THE DTID IN THE ABORT MESSAGE THE SAME AS THE OTID IN THE CONTINUE MESSAGE? CHECK D: WAS THE P-ABORT CAUSE IN THE ABORT MESSAGE THE CORRECT VALUE, (UNRECOGNIZED TRANSACTION ID)? 7. 8. CHECK E: WAS THE END MESSAGE CORRECTLY RECEIVED BY SP A? 9. CHECK F: WERE TSL STATE MACHINES ASSOCIATED WITH THESE TRANSACTIONS LEFT IN THE IDLE STATE AT SP A?

TEST NUMBER: 1.4.2 1) Sheet: 3 of 3

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets

Originating transaction ID value: OCTET STRING (1-4 octets long) X

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) Y

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) Z

(Not equal to X)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001B

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) Y

(OTID value received in CONTINUE message)

P-Abort cause tag: 01001010 P-Abort cause length: one octet

P-Abort cause value: 00000001 Unrecognized Transaction ID

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) X

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

TEST NUMBER: 1.4.2 2)		Sheet: 1 of 3
REFERENCE: 3.3.3.2/Q.774		
TITLE: Multiple Transaction Encoding	Inopportune Messages	
SUBTITLE: Message with unassigned I	OTID after transaction establishment	
PURPOSE: To verify that the signalling establishment of another trans	point A is able to correctly react to a Continue nsaction	message with DTID unassigned after the
PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPECTED MESSAGE SEQUENCE:		
SP A (TSL)		SP B (TSL)
TR-BEGIN req. =====>		
BEGIN	>	
	<	CONTINUE
TR-CONTINUE ind.		
	<	CONTINUE (new transaction)
ABORT (P)	>	
TR-P-ABORT ind. <======		
	<	END
TR-END ind. <=======		
,		

TEST NUMBER: 1.4.2 2) Sheet: 2 of 3

TEST DESCRIPTION

- 1. Arrange for SP A to send a Begin message to SP B.
- 2. Arrange for SP B to send a Continue message in response to Begin message from SP A.
- 3. Arrange for SP B to send a Continue message with unassigned DTID to SP A.
- 4. Arrange for SP B to respond with an End message to the Begin message.
- CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT BY SP A? 5.
- CHECK B: WERE THE CONTINUE MESSAGES CORRECTLY RECEIVED BY SP A? 6.
- 7. CHECK C: WAS THE DTID IN THE ABORT MESSAGE THE SAME AS THE OTID IN THE SECOND CONTINUE MESSAGE?
- 8. CHECK D: WAS THE P-ABORT CAUSE IN THE ABORT MESSAGE THE CORRECT VALUE, (UNRECOGNIZED TRANSACTION ID)?
- 9 CHECK E: WAS THE END MESSAGE CORRECTLY RECEIVED BY SP A?
- CHECK F: WERE TSL STATE MACHINES ASSOCIATED WITH THESE TRANSACTIONS LEFT IN THE IDLE 10. STATE AT SP A?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

BEGIN

Message type tag: 01100010

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) W

Component portion tag: 01101100

Component portion length: correct number of octets

CONTINUE (1st)

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) X

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) W

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

TEST NUMBER: 1.4.2 2) Sheet: 3 of 3

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

CONTINUE (2nd)

Message type tag: 01100101

Message type length: correct number of octets

Originating transaction ID tag: 01001000

Originating transaction ID length: correct number of octets
Originating transaction ID value: OCTET STRING (1-4 octets long) Y

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) Z

(Not equal to W)

Component portion tag: 01101100

Component portion length: correct number of octets

ABORT (P)

Message type tag: 01100111

Message type length: correct number of octets

Destination transaction ID tag: 01001001B

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) Y

(OTID value received in 2nd CONTINUE message)

P-Abort cause tag: 01001010 P-Abort cause length: one octet

P-Abort cause value: 00000001 Unrecognized Transaction ID

END

Message type tag: 01100100

Message type length: correct number of octets

Destination transaction ID tag: 01001001

Destination transaction ID length: correct number of octets

Destination transaction ID value: OCTET STRING (1-4 octets long) W

(OTID value received in BEGIN message)

Component portion tag: 01101100

Component portion length: correct number of octets

7.2 TC Component Sublayer test specification

7.2.1 Guidance on performing component sublayer tests

- a) For all the tests, the phrase "... component with correct information" in the test description means that the detail values in the indicated component will be syntactically verified against the information listed in the check table for components within messages.
- b) In some tests, a check is required to verify that the Invocation State Machine has returned to idle. One possible procedure to perform this check is to send a Return Result-Last component with the presumed idled Invoke ID. If the IUT (Implementation Under Test) returns a Reject with problem code = "unrecognized Invoke ID," the IUT has passed this check.
- c) For all tests of the CSL, the component has to be carried in a TSL message, e.g. the Invoke component in Test No. 2.1.1.1 is carried from SP A to SP B in a Begin message and the Return Result-Last component is carried in an End message. In fact, if a transaction is first established between SP A and SP B, it is possible to carry the Invoke and the Return Result components in Continue messages.
- d) The assumption used in these CSL tests is that the transaction is kept alive until the last component in the message flow has been delivered to the peer. In case this assumption does not hold for a real application (e.g. because of the use of an Abort or End message), one cannot reach any conclusive verdict on the test.
- e) CSL tests assume that the TSL and SCCP operate correctly. Thus, CSL tests assume that, in particular, components are carried in valid TSL messages within valid transaction states so that abnormal occurrences in the underlying (sub) layer(s) do not occur.
- f) TC-User related information, such as specific operation code and parameters, are not specified. It is up to the test implementers to include application dependent information, where applicable, in order to provoke the expected component flow.

7.2.2 Component sublayer test list

All tests are validation tests

Tests marked "*" are compatibility tests

2 Component Sublayer

2.1	Valid Functions
4.1	vanu i uncuons

	2.1.1	Invoke component, unlinked operations	
		2.1.1.1	Class 1 single operation invocation
*			2.1.1.1.1 IUT as sender: receive result
*			2.1.1.1.2 IUT as receiver: report result
*			2.1.1.1.3 IUT as sender: receive error
*			2.1.1.1.4 IUT as receiver: report error
*			2.1.1.1.5 IUT as sender: timer expiry
		2.1.1.2	Class 2 single operation invocation
*			2.1.1.2.1 IUT as sender: receive error
*			2.1.1.2.2 IUT as sender: timer expiry
		2.1.1.3	Class 3 single operation invocation
*			2.1.1.3.1 IUT as sender: receive result
*			2.1.1.3.2 IUT as sender: timer expiry
		2.1.1.4	Class 4 single operation invocation
*			2.1.1.4.1 IUT as sender

		Super	rseaea	by a more recent version
	2.1.2	Invoke com	nponent, linke	d operations
		2.1.2.1	-	inal operation invocation
*			2.1.2.1.1	IUT as sender: receive a linked Class 1 operation invocation, report result
*			2.1.2.1.2	IUT as receiver: send a linked Class 1 operation invocation, receive result
*			2.1.2.1.3	IUT as sender: receive a linked Class 1 operation invocation, report error
*			2.1.2.1.4	IUT as receiver: send a linked Class 1 operation invocation, receive error
		2.1.2.2		inal operation invocation
*		2.1.2.2	2.1.2.2.1	IUT as sender: receive a linked Class 2 operation invocation, no outcome
*			2.1.2.2.2	IUT as receiver: send a linked Class 2 operation invocation, timer expiry
	2.1.3	Remote Re		To T as receiver, some a miner class 2 operation invocation, timer expiry
	2.11.5	2.1.3.1	Remote Reje	ect by CSL
		2.1.5.1	2.1.3.1.1	General problem code
			2.1.3.1.2	Invoke problem code
			2.1.3.1.3	Return Result problem code
			2.1.3.1.4	Return Error problem code
		2.1.3.2		ect by TC-User
			2.1.3.2.1	Invoke problem code
			2.1.3.2.2	Return Result problem code
			2.1.3.2.3	Return Error problem code
		2.1.3.3		ect with an Invoke problem code
			2.1.3.3.1	Class 1 operation invocation
			2.1.3.3.2	Class 2 operation invocation
			2.1.3.3.3	Class 3 operation invocation
			2.1.3.3.4	Class 4 operation invocation
	2.1.4	Reception of	of component	leading to TC-User reject
		2.1.4.1	Invoke prob	
			2.1.4.1.1	Unrecognized operation code
			2.1.4.1.2	Unexpected linked operation
			2.1.4.1.3	Linked response unexpected
			2.1.4.1.4	Wrong type parameter
		2.1.4.2	Return Resu	ılt problem
			2.1.4.2.1	Wrong type parameter
		2.1.4.3	Return Error	r problem
			2.1.4.3.1	Unrecognized error
			2.1.4.3.2	Unexpected error
			2.1.4.3.3	Wrong type parameter
	2.1.5	Segmentati	on for Return	
		2.1.5.1		le operation invocation
*			2.1.5.1.1	IUT as sender: receive segmented components
*			2.1.5.1.2	IUT as receiver: send segmented components
		2.1.5.2	_	le operation invocation
*			2.1.5.2.1	IUT as sender: receive segmented component
	2.1.6	User Cance		
	2.1.7	Encoding V		
		2.1.7.1		length definite short
		2.1.7.2		length definite long
		2.1.7.3	•	length indefinite
		2.1.7.4	Value variat	
			2.1.7.4.1	Invoke ID
				2.1.7.4.1.1 Invoke ID = -127 (FFh)
				2.1.7.4.1.2 Invoke ID = 0 (00h)
			2.1.7.4.2	Global operation code
	2.1.8	_	mponents gro	
		2.1.8.1		erations invocation; receiving success
		2.1.8.2		erations invocation; reporting success
		2.1.8.3	A malforme	d component received

2.2	Syntacti	cally Inval	id Behavior			
	2.2.1	Invalid v	alues for info	ormation elements		
		2.2.1.1	Length of	Invoke ID > 1 in Invoke component		
		2.2.1.2		Invoke $ID = 0$ in Invoke component		
	2.2.2	Invalid st		•		
		2.2.2.1	Invoke cor	mponent		
			2.2.2.1.1			
			2.2.2.1.2	Operation code missing		
		2.2.2.2	Return Res	sult component		
			2.2.2.2.1	Invoke ID missing		
			2.2.2.2.2	Operation code missing while parameters included		
			2.2.2.2.2	Sequence tag missing while parameters included		
		2.2.2.3	Return Err	ror		
			2.2.2.3.1	Invoke ID missing		
			2.2.2.3.2	Error code missing		
		2.2.2.4	Unknown	component type		
			2.2.2.4.1	Invoke ID unrecognizable		
			2.2.2.4.2	invoke ID derivable		
	2.2.3	Invalid encoding for Invoke component				
			2.2.3.1	Invalid tag		
			2.2.3.2	Wrong component length		
			2.2.3.3	Missing EOC in indefinite form		
2.3	Inoppor	Inopportune behavior				
	2.3.1 Inopportune Invoke component			omponent		
		2.3.1.1	Invalid lin	ked ID		
	2.3.2	Unrecogn	nized Invoke	ID		
		2.3.2.1	Inopportur	ne Return Result-Last component		
		2.3.2.2	Inopportur	ne Return Result Not-Last component		
		2.3.2.3	Inopportur	ne Return Error component		
		2.3.2.4	Inopportur	ne Reject component		
	2.3.3	Unexpec	ted Compone	ents		
		2.3.3.1	Return Res	sult-Last for Class 2		
		2.3.3.2	Return Res	sult-Last for Class 4		
		2.3.3.3	Return Res	sult Not-Last for Class 2		
		2.3.3.4		sult Not-Last for Class 4		
		2.3.3.5		for for Class 3		
		2.3.3.6	Return Err	or for Class 4		

TEST NUMBER:	2.1.1.1.1		Sheet: 1 of 2	
REFERENCE: 3.2.1/Q.774				
TITLE: Valid Fun	ctions; Invoke compo	nent, unlinked operations		
SUBTITLE: Class	1 single operation in	vocation; IUT as sender: receive result		
PURPOSE: To ver can be	ify that a single Class received and delivere	s 1 operation can be successfully invoked and the to the TC-User	ne successful completion of the operation	
PRE-TEST CONDI	TIONS:			
1) Arrange the TC-	User stimulus such th	nat an appropriate TSL message generated at SP	A contains an Invoke component	
2) Arrange the data	a at SP B such that a F	Return Result-Last component can be generated		
CONFIGUR	ATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPECTED MESS	AGE AND COMPOR	NENT FLOW:		
SP A	(CSL)		SP B (CSL)	
TC-INVO	KE req. ====>			
INVOKE	(i)	>		
		<	RETURN RESULT-LAST (i)	
TC-RESU	LT-L ind.			
<=====				
TEST DESCRIPTION	ON			
1. Initiate a sir	ngle operation invocat	cion from SP A to SP B.		
2. CHECK A:	WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
	3. CHECK B: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?			
4. CHECK C:	4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component por Component por	Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP A to SP B				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.1.1.1.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-user)

RETURN RESULT-LAST component in TSL messages from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

NOTE - Omitted when no parameter is present

TEST	TEST NUMBER: 2.1.1.1.2 Sheet: 1 of 2			Sheet: 1 of 2
REFI	ERENCE: 3.2	.1/Q.774		
TITL	E: Valid Fund	ctions; Invoke compone	ent, unlinked operations	
SUB	TITLE: Class	1 single operation invo	ocation; IUT as receiver: report result	
PURI	POSE: To versent	ify that a Class 1 opera	ation can be successfully invoked and the successfully	ccessful completion of the operation can be
PRE-	TEST CONDI	TIONS: Arrange the Invoke comp	TC-User stimulus such that an appropriate TS onent	L message generated at SP B contains an
	CONFIGUR	ATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP
EXPI	ECTED MESS	AGE AND COMPON	ENT FLOW:	
	SP A (CSI	L)		SP B (CSL)
			<	INVOKE (i)
	TC-INVOKE i			
	<=====================================			
	TC-RESULT-I			
	RETURN-RI	ESULT-LAST (i)	>	
		,		
TEST	Γ DESCRIPTIO	ON		
1.	Initiate a sin	gle operation invocation	on from SP B to SP A.	
2.	2. CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?			
3.	3. CHECK B: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
4.	4. CHECK C: WAS THE INVOKE ID IN THE RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT?			MPONENT THE SAME AS THE ONE IN
5.	5. CHECK D: WAS THE OPERATION CODE IN THE RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVO	INVOKE component in TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.1.1.1.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL messages from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

NOTE - Omitted when no parameter is present

TEST NUMBER: 2.1.1.1.3		Sheet: 1 of 2	
REFERENCE: 3.2.1/Q.774			
TITLE: Valid Functions; Invoke compo	nent, unlinked operations		
SUBTITLE: Class 1 single operation in	vocation; IUT as sender: receive error		
PURPOSE: To verify that a Class 1 ope be received and delivered to	ration can be successfully invoked and the unsu	accessful completion of the operation can	
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such the 2) Arrange the data at SP B such that a R	at an appropriate TSL message generated at SP at the second sector of the sector of th	A contains an Invoke component	
CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPON	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	>		
	<	RETURN ERROR (i)	
TC-U-ERROR ind.			
TEST DESCRIPTION			
Initiate a single operation invocat	ion from SP A to SP B.		
2. CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3. CHECK B: WAS THE RETUR BY SP A?	N ERROR COMPONENT WITH CORRECT I	NFORMATION PASSED TO TC-USER	
4. CHECK C: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?		
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES		
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP A to SP B			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.1.1.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL messages from SP B to SP A

Component type tag: 10100011 (RETURN ERROR) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y (y is a valid error code)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.1.1.4			Sheet: 1 of 2		
REFE	REFERENCE: 3.2.1/Q.774				
TITLE	E: Valid Functions; Invoke compo	nent, unlinked operations			
SUBT	TTLE: Class 1 single operation in	vocation; IUT as receiver: report error			
PURP	OSE: To verify that a Class 1 ope be sent	eration can be successfully invoked and the unsu	accessful completion of the operation can		
1) Ar		nat an appropriate TSL message generated at SP lat a Return-Error component can be generated	B contains an Invoke component		
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP		
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:			
	SP A (CSL)		SP B (CSL)		
		<	INVOKE (i)		
	TC-U-ERROR ind.				
	`				
	TC-RESULT-L req. =====>				
	RETURN-ERROR (i)	>			
TEST	DESCRIPTION				
1.	Initiate a single operation invocat	ion from SP B to SP A.			
2.	2. CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?				
3.	CHECK B: WAS THE RETUR	N ERROR COMPONENT WITH CORRECT I	NFORMATION SENT BY BY SP A?		
4.	CHECK C: WAS THE INVOK INVOKE COMPON	E ID IN THE RETURN ERROR COMPONEN	NT THE SAME AS THE ONE IN THE		
5.	CHECK D: WAS THE ERROR	CODE IN THE RETURN ERROR COMPONE	ENT VALID?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES					
Component portion in TSL messages					
Component portion tag: 01101100 Component portion length: correct number of octets					
INVOKE component in TSL message from SP B to SP A					
Co Co	omponent type tag: 10100001 (IN omponent length: correct number o	VOKE) of octets			
In	voke ID tag: 00000010 voke ID length: 00000001 (one oc voke ID: i (i represents an integer)	tet)			

TEST NUMBER: 2.1.1.1.4 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR COMPONENT in TSL message from SP A to SP B

Component type tag: 10100011 (RETURN ERROR) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y

parameters (provided by the TC-User)

TEST NUMBER: 2.1.1.1.5 Sheet: 1 of 1

REFERENCE: 3.2.1/Q.774

TITLE: Valid Functions; Invoke component, unlinked operations

SUBTITLE: Class 1 single operation invocation; IUT as sender: timer expiry

PURPOSE: To verify that a Class 1 operation can be successfully invoked and the timer expiry indication can be delivered to the

TC-User

PRE-TEST CONDITIONS:

1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component

2) Arrange the data at SP B such that no component can be generated

CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP

EXPECTED MESSAGE AND COMPONENT FLOW:

SP A (CSL) SP B (CSL)

TC-INVOKE req.

INVOKE (i). ------

timer expiry for invocation (i)

TC-L-CANCEL ind. (i)

TEST DESCRIPTION

1. Initiate a single operation invocation from SP A to SP B.

2. CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?

3. CHECK B: WAS THE COMPONENT FLOW AS SHOWN ABOVE?

4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.1.2.1	TEST NUMBER: 2.1.1.2.1			
REFERENCE: 3.2.1/Q.774				
TITLE: Valid Functions; Invoke compo	nent, unlinked operations			
SUBTITLE: Class 2 single operation inv	vocation; IUT as sender: receive error			
PURPOSE: To verify that a Class 2 oper the TC-User	ration can be successfully invoked and the failur	e report can be received and delivered to		
PRE-TEST CONDITIONS:				
1) Arrange the TC-User stimulus such th	at an appropriate TSL message generated at SP	A contains an Invoke component		
2) Arrange the data at SP B such that a R	Return Error component can be generated			
CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPON	NENT FLOW:			
SP A (CSL)		SP B (CSL)		
TC-INVOKE req.				
INVOKE (i)	>			
(4)				
	<	RETURN ERROR (i)		
TC-U-ERROR ind.				
\				
TEST DESCRIPTION				
Initiate a single Class 2 operation	invocation from SP A to SP B.			
2. CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?		
3. CHECK B: WAS THE RETUR BY SP A?	N ERROR COMPONENT WITH CORRECT II	NFORMATION PASSED TO TC-USER		
4. CHECK C: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct no	umber of octets			
INVOKE component in TSL message fro	om SP A to SP B			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.1.1.2.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR COMPONENT in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN ERROR) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y

parameters (provided by the TC-User)

Sheet: 1 of 1 TEST NUMBER: 2.1.1.2.2 REFERENCE: 3.2.1/Q.774 TITLE: Valid Functions; Invoke component, unlinked operations SUBTITLE: Class 2 single operation invocation; IUT as sender: timer expiry PURPOSE: To verify that a Class 2 operation can be successfully invoked and the timer expiry indication can be delivered to the TC-User PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that no component will be generated CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP EXPECTED MESSAGE AND COMPONENT FLOW: SP A (CSL) SP B (CSL) TC-INVOKE req. INVOKE (i) timer expiry for invocation (i) TC-L-CANCEL ind. TEST DESCRIPTION 1. Initiate a single Class 2 operation invocation from SP A to SP B. CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? 2.. CHECK B: WAS THE COMPONENT FLOW AS SHOWN ABOVE? 3. CHECK C: WAS THE TC-USER AT SP A INFORMED OF TIMER EXPIRY? 4. 5. CHECK D: WAS THE INVOCATION STATE MACHINE IDLE AT SP A? CHECK TABLE FOR COMPONENTS WITHIN MESSAGES Component portion in TSL messages Component portion tag: 01101100 Component portion length: correct number of octets INVOKE component in TSL message from SP A to SP B Component type tag: 10100001 (INVOKE) Component length: correct number of octets Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.1.3.1			Sheet: 1 of 2		
REFE	REFERENCE: 3.2.1/Q.774				
TITLE	E: Valid Functions; Invoke compo	nent, unlinked operations			
SUBT	ITLE: Class 3 single operation in	vocation; IUT as sender: receive result			
PURPOSE: To verify that a single Class 3 operation can be successfully invoked and the successful report of the operation can be received and delivered to the TC-User					
PRE-T	TEST CONDITIONS:				
		at an appropriate TSL message generated at SP	A contains an Invoke component		
2) Ar	range the data at SP B such that a F	Return Result-Last component can be generated			
CONFIGURATION: 1 TYPE OF TEST: VAT and CI		TYPE OF TEST: VAT and CPT	TYPE OF SP: SP		
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:			
	SP A (CSL)		SP B (CSL)		
	TC-INVOKE req.				
	INVOKE (i)	>			
		<	RETURN RESULT-LAST (i)		
	TC-RESULT-L ind.				
TEST	DESCRIPTION				
1.	Initiate a single Class 3 operation invocation from SP A to SP B.				
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?				
3.	CHECK B: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?				
4.	4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?				
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES					
Component portion in TSL messages					
Component portion tag: 01101100 Component portion length: correct number of octets					
INVOKE component in TSL message from SP A to SP B					
Component type tag: 10100001 (INVOKE) Component length: correct number of octets					
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)					

TEST NUMBER: 2.1.1.3.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

NOTE - Omitted when no parameter is present

TEST NUMBER: 2.1.1.3.2 Sheet: 1 of 1 REFERENCE: 3.2.1/Q.774 TITLE: Valid Functions; Invoke component, unlinked operations SUBTITLE: Class 3 single operation invocation; IUT as sender: timer expiry PURPOSE: To verify that a Class 3 operation can be successfully invoked and the timer expiry indication can be delivered to the TC-User PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SPA contains an Invoke component 2) Arrange the data at SP B such that no component will be generated CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP EXPECTED MESSAGE AND COMPONENT FLOW: SP A (CSL) SP B (CSL) TC-INVOKE req. **INVOKE (i)** timer expiry for invocation (i) TC-L-CANCEL ind. TEST DESCRIPTION 1. Initiate a Class 3 operation invocation from SP A to SP B. 2. CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? CHECK B: WAS THE COMPONENT FLOW AS SHOWN ABOVE? 3. CHECK C: WAS THE TC-USER AT SP A INFORMED OF TIMER EXPIRY? 4. CHECK D: WAS THE INVOCATION STATE MACHINE IDLE AT SP A? CHECK TABLE FOR COMPONENTS WITHIN MESSAGES Component portion in TSL messages Component portion tag: 01101100 Component portion length: correct number of octets INVOKE component in TSL message from SP A to SP B Component type tag: 10100001 (INVOKE) Component length: correct number of octets Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer) Operation code tag: 00000010 (local) or 00000110 (global) Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.1.4.1 Sheet: 1 of 1

REFERENCE: 3.2.1/Q.774

TITLE: Valid Functions; Invoke component, unlinked operations

SUBTITLE: Class 4 single operation invocation; IUT as sender

PURPOSE: To verify that a Class 4 operation can be successfully initiated and no response is received.

PRE-TEST CONDITIONS: Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an

Invoke component

CONFIGURATION: 1 TYPE OF TEST: VAT and CPT TYPE OF SP: SP

EXPECTED MESSAGE AND COMPONENT FLOW:

SP A (CSL) SP B (CSL)

TC-INVOKE req.

INVOKE (i)

timer expiry for invocation (i)

TC-L-CANCEL ind.

TEST DESCRIPTION

1. Initiate a single Class 4 operation invocation from SP A to SP B.

2. CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?

3. CHECK B: WAS THE TC-USER AT SP A INFORMED OF TIMER EXPIRY?

4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.2.1.1				Sheet: 1 of 3	
REFE	ERENCE: 3.2.	.1/Q.774			
TITL	E: Valid Fund	ctions; Invoke compone	ent, Linked operations		
SUB	ΓΙΤLE: Class	1 original operation in	vocation; IUT as sender: receive a linked Class	s 1 operation invocation, report result	
PURI	POSE: To ver operation	rify that a linked Class on can be performed	1 operation can be successfully received and	the successful completion of the original	
PRE-	TEST CONDI	ΓΙΟΝS			
· ·	_		t an appropriate TSL message generated at SP ked Invoke component can be generated	A contains an Invoke component	
	CONFIGUR	ATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPI	ECTED MESS	AGE AND COMPONI	ENT FLOW:		
	SP A (CSI	ــ)		SP B (CSL)	
	TC-INVOKE r	req.			
	INVOKE (i)		>		
	II (VOILL (I)		<	INVOKE (j, i)	
	TC-INVOKE i	nd.			
	<==========	===			
	TC-RESULT-I	•			
	RETURN-RE	SULT-LAST (j)	>		
		(J)	<	RETURN-RESULT-LAST (i)	
	TC-RESULT-I	L ind.			
	<======				
TEST	DESCRIPTIO	ON			
1.	Initiate a linl	nked operation invocation from SP A to SP B.			
2.		WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3.	CHECK B:	WAS A LINKED INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?			
4.	CHECK C:	: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
5.	CHECK D:	WAS THE INVOKE ID IN THE RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT SENT BY SP B ?			
6.	CHECK E:	WAS THE OPERATION CODE IN THE RETURN RESULT-LAST COMPONENT SENT BY SP A THE SAME AS THE ONE IN THE INVOKE COMPONENT SENT BY SP B ?			
7.	CHECK F:	: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?			
8.	CHECK G:	CK G: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?			

TEST NUMBER: 2.1.2.1.1 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

INVOKE component in TSL message sent by SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in 2nd TSL message sent by SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

TEST NUMBER: 2.1.2.1.1 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long) (see Note)

Operation code: y (see Note)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

NOTE - Omitted when no parameter is present

TEST	NUMBER:	2.1.2.1.2	Sheet: 1 of 3		
REFE	ERENCE: 3.2	.1/Q.774			
TITL	E: Valid Fun	ctions; Invoke compo	nent, Linked operations		
SUBT	ΓΙΤLE: Class	1 original operation i	invocation; IUT as receiver: send a linked Class	1 operation invocation, receive result	
PURI		ify that a linked Clas on can be performed	s 1 operation can be successfully invoked and	the successful completion of the original	
PRE-	TEST CONDI		e TC-User stimulus such that an appropriate TSI ponent which will invoke a linked operation	L message generated at SP B contains an	
	CONFIGUR	ATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPE	ECTED MESS	AGE AND COMPO	NENT FLOW:		
	SP A (CS)	L)		SP B (CSL)	
			<	INVOKE (i)	
	TC-INVOKE				
	TC-INVOKE	•			
	INVOKE (j, i	i)	>		
			<	RETURN RESULT-LAST (j)	
	TC-RESULT-	L ind.			
	<=====	===			
	TC-RESULT-	L req.			
	RETURN-RI	ESULT-LAST (i)	>		
TEST DESCRIPTION					
Initiate a linked operation invocation from SP B to SP A.					
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?				
3.	CHECK B:	WAS A LINKED I	NVOKE COMPONENT WITH CORRECT INF	FORMATION SENT BY SP A?	
4.	CHECK C: WAS THE LINKED ID THE SAME AS THE ORIGINAL INVOKE ID SENT BY SP B?				
5.	CHECK D: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?				
6.	CHECK E: WAS THE SECOND RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION SENT BY SP A?				
7.	CHECK F: WAS THE INVOKE ID IN THE SECOND RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE ORIGINAL INVOKE COMPONENT SENT BY SP B?				
8.	CHECK G: WAS THE OPERATION CODE IN THE SECOND RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE ORIGINAL INVOKE COMPONENT?				
9.	CHECK H: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?				

TEST NUMBER: 2.1.2.1.2 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in initial TSL message from SP B to SP A

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

INVOKE component in TSL message sent from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message sent by SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

TEST NUMBER: 2.1.2.1.2 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long) (see Note)

Operation code: y (see Note)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message sent by SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

NOTE - Omitted when no parameter is present

TEST NUMBER: 2.1.2.1.3			Sheet: 1 of 3		
REFI	ERENCE: 3.2	1/Q.774			
TITL	E: Valid Fund	ctions; Invoke compo	nent, Linked operations		
SUB	ΓΙΤLE: Class	1 original operation i	nvocation; IUT as sender: receive a linked Class	s 1 operation invocation, report error	
PURPOSE: To verify that a linked Class 1 operation can be successfully received and the reporting error will not impact the completion of the original operation					
PRE-	TEST CONDI	TIONS:			
	-		nat an appropriate TSL message generated at SP inked invocation can be generated	A contains an Invoke component	
CONFIGURATION: 1			TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPI	ECTED MESS	AGE AND COMPO	NENT FLOW:		
SP A (CSL)				SP B (CSL)	
	TC-INVOKE	req. ===>			
	INVOKE (i)		>		
			<	INVOKE (j, i)	
	TC-INVOKE i				
	TC-U-ERROR	1			
RETURN-ERROR (j)		RROR (j)	>		
	TC DECLUTE	T · 1	<	RETURN-RESULT-LAST (i)	
TC-RESULT-L ind. <====================================					
TEST DESCRIPTION					
1.	Initiate a linked operation invocation from SP A to SP B.				
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?				
3.	CHECK B:	CHECK B: WAS A LINKED INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?			
4.	CHECK C:	CK C: WAS THE RETURN ERROR COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
5.	CHECK D:	CHECK D: WAS THE INVOKE ID IN THE RETURN ERROR COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT SENT BY SP B ?			
6.	CHECK E: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?				
7.	CHECK F: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?				

TEST NUMBER: 2.1.2.1.3 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

INVOKE component in the TSL messages sent by SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in the TSL message sent by SP A

Component type tag: 10100010 (RETURN ERROR)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

TEST NUMBER: 2.1.2.1.3 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Error code tag: 00000010 (local) or 00000110 (global) (see Note)

Error code length: correct number of octets (e.g. 00000001 if z is one octet long) (see Note)

Error code: z (see Note)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in the TSL message sent by SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

NOTE - Omitted when no parameter is present

TEST	EST NUMBER: 2.1.2.1.4			Sheet: 1 of 3	
REFE	ERENCE: 3.2	.1/Q.774			
TITL	E: Valid Fund	ctions; Invoke compo	onent, Linked operations		
SUB	ΓΙΤLE: Class	1 original operation	invocation; IUT as receiver: send a linked Class	1 operation invocation, receive error	
PURI		ify that a linked Cla tion of the original o	ss 1 operation can be successfully invoked and peration	I the receiving error will not impact the	
PRE-	TEST CONDI	TIONS: Arrange the Invoke con	e TC-User stimulus such that an appropriate TSL aponent	message generated at SP B contains an	
	CONFIGUR	ATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPI	ECTED MESS	AGE AND COMPO	NENT FLOW		
	SP A (CSI	L)		SP B (CSL)	
			<	INVOKE (i)	
	TC-INVOKE i				
	<======				
	TC-INVOKE 1	-			
	INVOKE (j, i)				
	< RETURN ERROR (j)				
	TC-U-ERROR ind.				
	<======	===			
	TC-RESULT-	1			
	RETURN-RI	ESULT-LAST (i)	>		
TEST	DESCRIPTION	ON			
1.	Initiate a lin	ked operation invoca	tion from SP B to SP A.		
2.	CHECK A:	WAS THE INVO BY SP A?	OKE COMPONENT WITH CORRECT INF	ORMATION PASSED TO TC-USER	
3.	CHECK B:	WAS A LINKED I	NVOKE COMPONENT WITH CORRECT INF	CORMATION SENT BY SP A?	
4.	CHECK C: WAS THE LINKED ID THE SAME AS THE ORIGINAL INVOKE ID SENT BY SP B?			E ID SENT BY SP B?	
5.	CHECK D: WAS THE RETURN ERROR COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?		NFORMATION PASSED TO TC-USER		
6.	CHECK E:	WAS THE RET BY SP A?	URN RESULT-LAST COMPONENT WITH	I CORRECT INFORMATION SENT	
7.	7. CHECK F: WAS THE INVOKE ID IN THE RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE ORIGINAL INVOKE COMPONENT SENT BY SP B?				
8.	CHECK G: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?				

TEST NUMBER: 2.1.2.1.4 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message S from SP B to SP A

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

INVOKE component in TSL messages by SP A

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message sent by SP B

Component type tag: 10100011 (RETURN ERROR)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

TEST NUMBER: 2.1.2.1.4 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if z is one octet long)

Error code: z

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message sent by SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

NOTE - Omitted when no parameter is present

TEST	TEST NUMBER: 2.1.2.2.1		Sheet: 1 of 2	
REFE	RENCE: 3.2.1/Q.774			
TITLE	E: Valid Functions; Invoke compo	nent, Linked operations		
SUBT	ITLE: Class 4 original operation is	nvocation; IUT as sender: receive a linked Class	2 operation invocation, no outcome	
PURP	OSE: To verify that a linked Class Class 4 operation can be per	s 2 operation can be successfully received and formed	the successful completion of the original	
PRE-T	TEST CONDITIONS:			
		at an appropriate TSL message generated at SP named Class 2 Invoke component can be generated.		
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPON	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
,	TC-INVOKE req.			
]	INVOKE (i)	>		
	INVOKE (j, i)			
	TC-INVOKE ind.			
1	timer expiry for invocation (i)			
TC-L-CANCEL ind. <====================================				
TEST	DESCRIPTION			
1.	Initiate a linked operation invocat	ion from SP A to SP B.		
2.	CHECK A: WAS THE INVOKE	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3.	CHECK B: WAS A LINKED I BY SP A?	NVOKE COMPONENT WITH CORRECT IN	NFORMATION PASSED TO TC-USER	
4.	4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?			
CHEC	CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVO	KE component in TSL message fro	m SP A to SP B		
Co Co	Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
In	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.2.2.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

INVOKE component in TSL message sent by SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.2.2.2		Sheet: 1 of 2		
REFERENCE: 3.2.1/Q.774				
TITLE: Valid Functions; Invoke compo	onent, Linked operations			
SUBTITLE: Class 4 original operation	invocation; IUT as receiver: send a linked Class	2 operation invocation, timer expiry		
PURPOSE: To verify that a linked Cla Class 4 operation can be pe	ss 2 operation can be successfully invoked and rformed	the successful completion of the original		
PRE-TEST CONDITIONS: Arrange the Invoke con	e TC-User stimulus such that an appropriate TSL aponent which will invoke a Class 2 linked opera	message generated at SP B contains an tion		
CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPO	NENT FLOW:			
SP A (CSL)		SP B (CSL)		
	<	INVOKE (i)		
TC-INVOKE ind.				
<=========				
TC-INVOKE req. ======>				
INVOKE (j, i) timer expiry for invocation (j)	>			
TC-L-CANCEL ind.				
TEST DESCRIPTION				
Initiate a linked operation invoca	ntion from SP B to SP A.			
2. CHECK A: WAS THE INVOK BY SP A?				
3. CHECK B: WAS A LINKED	3. CHECK B: WAS A LINKED INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
4. CHECK C: WAS THE LINKE	D ID THE SAME AS THE ORIGINAL INVOK	E ID SENT BY SP B?		
5. CHECK D: WAS THE INVOC	CATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES			
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in initial TSL mess	INVOKE component in initial TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.1.2.2.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

INVOKE component in TSL message sent from SP A to SP B

Component type tag: 10100010 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)

Invoke ID: j (j represents)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.3.1.1		Sheet: 1 of 2	
REFERENCES: 3.2.1/Q.774; 3.8.1/Q.7	72		
TITLE: Valid Functions; Remote Reject	1		
SUBTITLE: Remote Reject by CSL; Ge	eneral problem code		
PURPOSE: To verify that a remote reject	tion by CSL with general problem code can be d	lelivered to the TC-User	
PRE-TEST CONDITIONS:			
	nat an appropriate TSL message generated at SP and appropriate TSL message generated at SP and appropriate code can be seen as a second control of the code can be seen as a second control of the code can be seen as a second control of the code can be seen as a second code can	=	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPOR	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	>		
3 31 <u>—</u> (4)	<	REJECT (i)	
TC-R-REJECT ind.			
TEST DESCRIPTION			
Initiate a single Class 1 operation	invocation from SP A to SP B.		
2. CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3. CHECK B: WAS THE REJE BY SP A?	CT COMPONENT WITH CORRECT INFO	ORMATION PASSED TO TC-USER	
4. CHECK C: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages	Component portion in TSL messages		
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from	om SP A to SP B		
Component type tag: 10100001 (IN Component length: correct number of	VOKE) of octets		
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.3.1.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL messages sent from SP B to SP A

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000000 (General Problem)

Problem code length: 00000001

Problem code: 00000000 (unrecognized component)

TEST NUMBER: 2.1.3.1.2		Sheet: 1 of 2
REFERENCES: 3.2.1/Q.774; 3.8.2/Q.7	72	
TITLE: Valid Functions; Remote Rejec	t	
SUBTITLE: Remote Reject by CSL; In	voke problem code	
PURPOSE: To verify that the remote re	jection by CSL with Invoke problem code can be	received and delivered to the TC-User
PRE-TEST CONDITIONS:		
	hat an appropriate TSL message generated at SP Reject component with Invoke problem code can	_
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPECTED MESSAGE AND COMPO	NENT FLOW:	
SP A (CSL)		SP B (CSL)
	<	INVOKE (i)
<i>TC-INVOKE ind.</i> ====>		
TC-INVOKE req.		
INVOKE (j,i)	>	
	<	REJECT (j)
TC-R-REJECT-L ind.		
TEST DESCRIPTION		
Initiate a linked Class 1 operation	n invocation from SP B to SP A.	
2. CHECK A: WAS THE INVO	DKE COMPONENT WITH CORRECT INF	ORMATION PASSED TO TC-USER
3. CHECK B: WAS THE REJE BY SP A?	ECT COMPONENT WITH CORRECT INFO	ORMATION PASSED TO TC-USER
4. CHECK C: WAS THE INVOC	CATION STATE MACHINE IDLE AT SP A?	
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES	
Component portion in TSL messages		
Component portion tag: 01101100 Component portion length: correct	number of octets	
INVOKE component in TSL message from	om SP B to SP A	
Component type tag: 10100001 (INVOKE) Component length: correct number of octets		
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)		

TEST NUMBER: 2.1.3.1.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents an operation code)

parameters (provided by the TC-User)

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)
Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL messages from SP B to SP A

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001

Problem code: 00000101 (unrecognized linked ID)

TEST NUMBER: 2.1.3.1.3		Sheet: 1 of 2		
REFERENCE: 3.2.1/Q.774; 3.8.3/Q.772	REFERENCE: 3.2.1/Q.774; 3.8.3/Q.772			
TITLE: Valid Functions; Remote Reject	ì			
SUBTITLE: Remote Reject by CLS; Re	turn Result problem code			
PURPOSE: To verify that a single Clas delivered to the TC-User	s 1 operation can be successfully invoked and	the remote rejection can be received and		
PRE-TEST CONDITIONS:				
1) Arrange the TC-User stimulus such th	at an appropriate TSL message generated at SP	B contains an Invoke component		
2) Arrange the data at SP B such that a F	Reject component can be generated			
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPO	NENT FLOW:			
SP A (CSL)		SP B (CSL)		
Si ii (CSE)		, ,		
	<	INVOKE (i)		
TC-INVOKE ind.				
TC-RESULT-L req.				
RETURN RESULT-LAST (i)	>			
TC-R-REJECT-ind. <======	<	REJECT (i)		
TEST DESCRIPTION				
Initiate a single Class 1 operation	invocation from SP B to SP A.			
2. CHECK A: WAS THE INVOI SP A?	KE COMPONENT WITH CORRECT INFOR	MATION PASSED TO TC-USER BY		
3. CHECK B: WAS THE REJECT SP A?	T COMPONENT WITH CORRECT INFORM	MATION PASSED TO TC-USER BY		
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES			
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from	om SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.1.3.1.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

Parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SPA

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000000 (unrecognized Invoke ID)

NOTE - Omitted when no parameter is present

TEST NUMBER: 2.1.3.1.4		Sheet: 1 of 2
REFERENCE: 3.2.1/Q.774; 3.8.4/Q.77	2	
TITLE: Valid Functions; Remote Rejec	t	
SUBTITLE: Remote Reject by CLS; Re	eturn Error problem code	
PURPOSE: To verify that a single Clas delivered to the TC-User	s 1 operation can be successfully invoked and	the remote rejection can be received and
PRE-TEST CONDITIONS:		
	nat an appropriate TSL message generated at SP	B contains an Invoke component
2) Arrange the data at SP B such that a I		b contains an invoke component
· · · · · ·		
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPECTED MESSAGE AND COMPO	NENT FLOW:	
SP A (CSL)		SP B (CSL)
	<	INVOKE (i)
	<u></u>	INVOKE (I)
TC-INVOKE ind. <======		
\		
TC-U-ERROR req.		
======>		
RETURN ERROR (i)	>	
	<	REJECT (i)
		RESECT (I)
TC-R-REJECT ind. <=======		
TEST DESCRIPTION		
Initiate a single Class 1 operation	invocation from SP B to SP A.	
2. CHECK A: WAS THE INVOI SP A?	KE COMPONENT WITH CORRECT INFORM	MATION PASSED TO TC-USER BY
3. CHECK B: WAS THE REJECT SP A?	CT COMPONENT WITH CORRECT INFORM	MATION PASSED TO TC-USER BY
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES	
Component portion in TSL messages		
Component portion tag: 01101100 Component portion length: correct number of octets		
INVOKE component in TSL message from	om SP B to SP A	
Component type tag: 10100001 (INVOKE) Component length: correct number of octets		
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)		

TEST NUMBER: 2.1.3.1.4 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP A to SP B

Component type tag: 10100011 (RETURN ERROR)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y

parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SPA

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000011 (RETURN ERROR)

Problem code length: 00000001

Problem code: 00000001 (unrecognized Invoke ID)

TEST	NUMBER: 2.1.3.2.1		Sheet: 1 of 2	
REFE	RENCE: 3.2.1/Q.774; 3.8.2/Q.77	2		
TITLI	E: Valid Functions; Remote Rejec	t		
SUBT	TTLE: Remote Reject by TC-Use	r; Invoke problem code		
PURP	OSE: To verify that the remote the TC-User	rejection by TC-User with Invoked problem	code can be received and delivered to	
1) Ar	PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that a Reject component can be generated			
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	>		
	(4)	<	DE LECT (:)	
	TC-U-REJECT ind.	\	REJECT (i)	
TEST	DESCRIPTION			
1.	Initiate a single Class 1 operation	n invocation from SP A to SP B.		
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3.	CHECK B: WAS THE REJECTION SP A?	CT COMPONENT WITH CORRECT INFORM	MATION PASSED TO TC-USER BY	
4.	CHECK C: WAS THE INVOC	CATION STATE MACHINE IDLE AT SP A?		
CHEC	CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Comp	Component portion in TSL messages			
Co Co	Component portion tag: 01101100 Component portion length: correct number of octets			
INVO	KE component in TSL message from	om SP A to SP B		
Co Co	Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
In	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.3.2.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents an operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SP A

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000001 (RETURN ERROR)

Problem code length: 00000001

Problem code: 00000000 (unrecognized Invoke ID)

TEST	NUMBER: 2.1.3.2.2		Sheet: 1 of 2
REFE	RENCE: 3.2.1/Q.774; 3.8.3/Q.772	2	
TITLI	E: Valid Functions; Remote Reject	1	
SUBT	TTLE: Remote Reject by TC-User	; Return Result problem code	
PURP	OSE: To verify that the remote rej TC-User	ection by TC-User with Return Result problem	code can be received and delivered to the
1) Ar	_	nat an appropriate TSL message generated at SP Reject component with Return Result problem co	•
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE AND COMPOR	NENT FLOW:	
	SP A (CSL)		SP B (CSL)
		<	INVOKE (i)
	TC-INVOKE ind.		
	TC-RESULT-L req.		
	RETURN RESULT-LAST (i)	>	
	TC-U-REJECT ind.	<	REJECT (i)
TEST	DESCRIPTION		
1.	Initiate a single Class 1 operation	invocation from SP B to SP A.	
2.	CHECK A: WAS THE INVOI SP A?	KE COMPONENT WITH CORRECT INFOR	MATION PASSED TO TC-USER BY
3.	CHECK B: WAS THE REJECT SP A?	T COMPONENT WITH CORRECT INFORM	MATION PASSED TO TC-USER BY
CHEC	CK TABLE FOR COMPONENTS	WITHIN MESSAGES	
Comp	onent portion in TSL messages		
Co Co	Component portion tag: 01101100 Component portion length: correct number of octets		
INVO	KE component in TSL message fro	om SP B to SP A	
Co Co	omponent type tag: 10100001 (IN omponent length: correct number of	VOKE) of octets	
In	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)		

TEST NUMBER: 2.1.3.2.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents an operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SP A

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000010 (wrong type parameter)

NOTE – Omitted when no parameter is present

TEST	NUMBER: 2.1.3.2.3		Sheet: 1 of 2
REFE	RENCE: 3.2.1/Q.774; 3.8.4/Q.772		
TITLI	E: Valid Functions; Remote Reject		
SUBT	TTLE: Remote Reject by TC-User,	, Return Error problem code	
PURP	OSE: To verify that the remote rej	ection by TC-User with Return Error problem of	code can be received and delivered to the
1) Ar	_	at an appropriate TSL message generated at SP i	•
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE AND COMPON	IENT FLOW:	
	SP A (CSL)		SP B (CSL)
		<	INVOKE (i)
	TC-INVOKE ind.		
	TC-U-ERROR req.		
	RETURN ERROR (i)	>	
		<	REJECT (i)
	TC-U-REJECT ind.		
TEST	DESCRIPTION		
1.	Initiate a single Class 1 operation	invocation from SP B to SP A.	
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFOR	MATION PASSED TO TC-USER BY
3.	CHECK B: WAS THE REJECT SP A?	T COMPONENT WITH CORRECT INFOR	MATION PASSED TO TC-USER BY
CHEC	CHECK TABLE FOR COMPONENTS WITHIN MESSAGES		
Comp	onent portion in TSL messages		
Co Co	Component portion tag: 01101100 Component portion length: correct number of octets		
INVO	KE component in TSL message from	m SP B to SP A	
Co Co	Component type tag: 10100001 (INVOKE) Component length: correct number of octets		
In	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)		

TEST NUMBER: 2.1.3.2.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents an operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP A to SP B

Component type tag: 10100011 (RETURN ERROR)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y

parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SPA

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000011 (RETURN ERROR)

Problem code length: 00000001

Problem code: 00000010 (unrecognized error)

TEST	NUMBER: 2.1.3.3.1		Sheet: 1 of 2	
REFE	RENCE: 3.2.1/Q.774			
TITLE	E: Valid Functions; Remote Rejec	t		
SUBT	ITLE: Remote Reject with an Inv	oke problem code; Class 1 operation invocation		
PURP	OSE: To verify that a single Clas delivered to the TC-User	s 1 operation can be successfully invoked and t	the remote rejection can be received and	
PRE-T	TEST CONDITIONS:			
	-	nat an appropriate TSL message generated at SP	A contains an Invoke component	
2) Ar	range the data at SP B such that a I	Reject component can be generated		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	>		
	I (V OKL (I)	<	REJECT (i)	
			RESECT (I)	
	TC-U-REJECT ind.			
TEST	DESCRIPTION			
1.	Initiate a single Class 1 operation	invocation from SP A to SP B.		
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3.	CHECK B: WAS THE REJECTION SP A?	CT COMPONENT WITH CORRECT INFOR	MATION PASSED TO TC-USER BY	
4.	CHECK C: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?		
CHEC	CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages				
Co Co	Component portion tag: 01101100 Component portion length: correct number of octets			
INVO	KE component in TSL message from	om SP A SP B		
Co Co	Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
In	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.3.3.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SPA

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001

Problem code: 00000010 (wrong type parameter)

TEST NUMBER: 2.1.3.3.2		Sheet: 1 of 2		
REFERENCE: 3.2.1/Q.774				
TITLE: Valid Functions; Remote Reject	zt			
SUBTITLE: Remote Reject with an In	voke problem code; Class 2 operation invocation			
PURPOSE: To verify that a single Cla delivered to the TC-User	ss 2 operation can be successfully invoked and t	the remote rejection can be received and		
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such t 2) Arrange the data at SP B such that a	that an appropriate TSL message generated at SP. Reject component can be generated	A contains an Invoke component		
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPO	NENT FLOW:			
SP A (CSL)		SP B (CSL)		
TC-INVOKE req.				
INVOKE (i)	>			
, ,	<	REJECT (i)		
TC-U-REJECT ind.				
<=======				
TEST DESCRIPTION				
Initiate a single Class 2 operation	n invocation from SP A to SP B.			
2. CHECK A: WAS THE INVOKE	KE COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?		
3. CHECK B: WAS THE REJE SP A?	CT COMPONENT WITH CORRECT INFOR	MATION PASSED TO TC-USER BY		
4. CHECK C: WAS THE INVOC	CATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS	CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages	Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP A SP B				
Component type tag: 10100001 (IN Component length: correct number	IVOKE) of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.1.3.3.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SPA

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001

Problem code: 00000010 (wrong type parameter)

TEST NUMBER: 2.1.3.3.3		Sheet: 1 of 2		
REFE	REFERENCE: 3.2.1/Q.774			
TITLE	E: Valid Functions; Remote Rejec	t		
SUBT	ITLE: Remote Reject with an Inv	oke problem code; Class 3 operation invocation		
PURP	OSE: To verify that a single Clas	s 3 operation can be successfully invoked and t	the remote rejection can be received and	
PRE-T	TEST CONDITIONS:			
1) Ar	range the TC-User stimulus such the	nat an appropriate TSL message generated at SP	A contains an Invoke component	
2) Ar	range the data at SP B such that a I	Reject component can be generated		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	>		
	IVORE (I)	<	REJECT (i)	
		<u></u>	RESECT (I)	
	TC-U-REJECT ind.			
TEST	DESCRIPTION			
1.	Initiate a single Class 3 operation	invocation from SP A to SP B.		
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3.	CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?			
4.	CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP A to SP B				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
In	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.3.3.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents an operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SPA

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001

Problem code: 00000010 (wrong type parameter)

TEST NUMBER: 2.1.3.3.4		Sheet: 1 of 2	
REFERENCE: 3.2.1/Q.774			
TITLE: Valid Functions; Remote Reject	et		
SUBTITLE: Remote Reject with an In-	voke problem code; Class 4 operation invocation		
PURPOSE: To verify that a single Class 4 operation can be successfully invoked and the remote rejection can be received and delivered to the TC-User			
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that a Reject component can be generated			
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPO	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	>		
	<	REJECT (i)	
TC-U-REJECT ind.			
<=======			
TEST DESCRIPTION			
Initiate a single Class 4 operation	n invocation from SP A to SP B.		
2. CHECK A: WAS THE INVOL	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?		
3. CHECK B: WAS THE REJE SP A?			
4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP A SP B			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.3.3.4 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP B to SPA

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001

Problem code: 00000010 (wrong type parameter)

TEST NUMBER: 2.1.4.1.1		Sheet: 1 of 2		
REFERENCE: 3.2.2.2/Q.774				
TITLE: Valid Functions; Reception of component le	ading to TC-User reject			
SUBTITLE: Invoke problem; Unrecognized operation	n code			
PURPOSE: To verify that a rejection of a requested of	peration can be performed			
PRE-TEST CONDITIONS: Arrange the stimulus such that an appropriate TSL message generated at SP B contains an Invoke component with an error as described below				
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPONENT FLOW	<i>!</i> :			
SP A (CSL)		SP B (CSL)		
<		INVOKE (i)		
TC-INVOKE ind.				
TC-U-REJECT req.				
REJECT (i)	>			
TEST DESCRIPTION				
1. Initiate an operation invocation from SP B to	SP A with an unrecognized operation	code.		
2. CHECK A: WAS THE REJECT COMPONE	CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3. CHECK B: WAS THE INVOKE ID IN THE COMPONENT?	E REJECT COMPONENT THE SA	AME AS THE ONE IN THE INVOKE		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP B to SP A				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.1.4.1.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents an invalid operation code)

parameters (provided by the TC-User)

REJECT component in TSL message sent from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000001 (INVOKE problem type)
Problem code length: 00000001

Problem code: 00000001 (unrecognized operation)

TEST NUMBER: 2.1.4.1.2			Sheet: 1 of 3	
REFE	REFERENCE: 3.2.2/Q.774			
TITLI	E: Valid Functions; Reception of c	omponent leading to TC-User reject		
SUBT	TTLE: Invoke problem; Unexpector	l linked operation		
PURP	OSE: To verify that a rejection ca	n be successfully initiated due to an unexpecte	d linked operation and without affecting	
PRE-	TEST CONDITIONS:			
		at an appropriate TSL message generated at SP.	A contains an Invoke component	
	_	Invoke with a linked ID is contained in an appro	-	
	-	11		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPON	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	>		
		<	INVOKE (j, i)	
	TC-INVOKE ind. <======			
	TC-U-REJECT req.			
	/			
	REJECT (j)	>		
		<	RETURN RESULT-LAST (i)	
			RETURN RESULT-LAST (I)	
	TC-RESULT-L ind.			
TEST DESCRIPTION				
1.	Initiate an unlinked operation invocation from SP A to SP B.			
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3.	CHECK B: WAS A LINKED INVOKE COMPONENT PASSED TO THE TC-USER BY SP A?			
4.	CHECK C: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
5.	CHECK D: WAS THE INVOKE ID IN THE REJECT COMPONENT THE SAME AS THE INVOKE ID IN THE INVOKE COMPONENT SENT BY SP B?			
6.	CHECK E: WAS THE INVOCA	ATION STATE MACHINE IDLE AT SP A?		

TEST NUMBER: 2.1.4.1.2 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100 Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

INVOKE component in the TSL message sent by SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i (i is an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents an operation code not linked to x)

parameters (provided by the TC-User)

REJECT component in TSL message sent by SP A

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001

Problem code: 00000111 (unexpected linked operation)

TEST NUMBER: 2.1.4.1.2 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: corect number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

NOTE - Omitted when no parameter is present

TEST NUMBER: 2.1.4.1.3			Sheet: 1 of 3	
REFE	REFERENCE: 3.2.1/Q.774			
TITLI	TITLE: Valid Functions; Reception of component leading to TC-User reject			
SUBT	TTLE: Invoke problem; Linked re	sponse unexpected		
PURP	OSE: To verify that an unexpected	l linked response can be rejected		
	TEST CONDITIONS:			
l) Ar wi	range the TC-User stimulus such the linvoke a linked operation	hat an appropriate TSL message generated at SP	B contains an Invoke component which	
	range the data at SP B such that a the operation	linked response contains at least one parameter	which is not associated with the outcome	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPOR	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	·····>	DIVOKE (L.)	
		<	INVOKE (j, i)	
	TC-INVOKE ind.			
	TC-U-REJECT req. ======>			
	REJECT (j)	>		
		<	RETURN RESULT-LAST (i)	
	TC-RESULT-L ind		()	
	<=====================================			
TEST DESCRIPTION				
1.	Initiate an operation invocation from SP A to SP B.			
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3.	CHECK B: WAS A LINKED INVOKE COMPONENT PASSED TO THE TC-USER BY SP A?			
4.	CHECK C: WAS THE REJECT COMPONENT SENT BY SP A?			
5.	CHECK D: WAS THE RETU TO TC-USER BY S	RN RESULT-LAST COMPONENT WITH SP A?	CORRECT INFORMATION PASSED	
6.	CHECK E: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?		

TEST NUMBER: 2.1.4.1.3 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)
Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code that does not allow any linked operation)

parameters (provided by the TC-User)

INVOKE component in the TSL message sent from SP B to SP A

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)
Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL message sent by SP A

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

TEST NUMBER: 2.1.4.1.3 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001

Problem code: 00000111 (linked response unexpected)

RETURN RESULT-LAST component in TSL message by SP B

Component type tag: 10100010 (RETURN RESULT-LAST) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: corect number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST	NUMBER: 2.1.4.1.4		Sheet: 1 of 2		
REFE	RENCE: 3.2.2.2/Q.774				
TITLI	E: Valid Functions; Reception of co	omponent leading to TC-User reject			
SUBT	ITLE: Invoke problem; Wrong typ	pe parameter			
PURP	OSE: To verify that a rejection of a	a requested operation can be performed			
PRE-T	TEST CONDITIONS: Arrange ti componer	he stimulus such that an appropriate TSL messant with an error as described below	age generated at SP B contains an Invoke		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPE	CTED MESSAGE AND COMPON	NENT FLOW:			
	SP A (CSL)		SP B (CSL)		
		<	INVOKE (i)		
	TC-INVOKE ind.				
	TC-U-REJECT req.				
	REJECT (i)	>			
	RESECT (I)				
TEST	DESCRIPTION				
1.	Initiate an operation invocation from	om SP B to SP A with a wrong type parameter i	ncluded.		
2.		COMPONENT WITH CORRECT INFORMA			
2	CHECK D. WAS THE DIVOK	E ID DUTTHE DELECT COMPONENT THE	NAME AS THE ONE DITTIE DIVOVE		
3.	CHECK B: WAS THE INVOK COMPONENT?	E ID IN THE REJECT COMPONENT THE S	SAME AS THE ONE IN THE INVOKE		
CHEC	CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages					
	Component portion tag: 01101100 Component portion length: correct number of octets				
INVO	INVOKE component in TSL message from SP B to SP A				
	Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
In	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.1.4.1.4 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User, including at least one parameter which is not one of those associated with the operation)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: i

Problem code tag: 10000001 (Invoke problem type)

Problem code length: 00000001

Problem code: 00000010 (wrong type parameter)

TEST	NUMBER: 2.1.4.2.1		Sheet: 1 of 2
REFE	RENCE: 3.2.2/Q.774		
TITLI	E: Valid Functions; Reception of c	omponent leading to TC-User reject	
SUBT	TITLE: Return Result problem; Wr	rong type parameter	
PURP	POSE: To verify that a rejection of Result-Last component	can be successfully initiated due to an invalid	l operation code included in the Return
1) Ar Cl	ass 1 or 3	that an appropriate TSL message generated at seturn Result-Last with an invalid operation code	_
2) 111	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXDE	CCTED MESSAGE AND COMPON		THE OF SI. SI
	SP A (CSL)	VENT LOW.	SP B (CSL)
	TC-INVOKE req.		
	INVOKE (i)	>	
		<	RETURN RESULT-LAST (i)
	TC-RESULT-L ind. <======		
	TC-U-REJECT req. =====>		
	REJECT (i)	>	
TEST	DESCRIPTION		
1.	Initiate an operation invocation fr Generate a response from SP B to	om SP A to SP B. SP A with a valid Invoke ID but a different ope	eration code.
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?
3.	CHECK B: WAS THE RETUR	N RESULT-LAST COMPONENT PASSED TO	O TC-USER BY SP A?
4.	CHECK C: WAS THE REJECT	T COMPONENT SENT BY SP A?	
5.	CHECK D: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?	

TEST NUMBER: 2.1.4.2.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long) (see Note)

Operation code: y (y is different from x) (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000010 (wrong type parameter)

TEST	NUMBER: 2.1.4.3.1		Sheet: 1 of 2	
REFE	CRENCE: 3.2.2/Q.774			
TITL	E: Valid Functions; Reception of co	omponent leading to TC-User reject		
SUBT	TITLE: Return Error problem; Unro	ecognized error		
PURF	POSE: To verify that a rejection car component	n be successfully initiated due to an unrecognize	d error code included in the Return Error	
1) Aı Cl	ass 1	that an appropriate TSL message generated at		
2) A1	range the data at SP B such that a R	eturn Error with an invalid error code is generat	ted	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	ECTED MESSAGE AND COMPON	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req. ====>			
	INVOKE (i)	>		
		<	RETURN ERROR (i)	
	TC-U-ERROR ind.			
	TC-U-REJECT req.			
	REJECT (i)	>		
TEST	TEST DESCRIPTION			
1.	Initiate A Class 1 operation invoc. Generate an unsuccessful respons	ation from SP A to SP B. e from SP B to SP A with a valid Invoke ID but	an invalid error code for this operation	
2.	-	E COMPONENT WITH CORRECT INFORMA	-	
3.	CHECK B: WAS THE RETUR	N ERROR COMPONENT PASSED TO TC-US	SER BY SP A?	
4.	CHECK C: WAS THE REJECT	COMPONENT SENT BY SP A?		
5.	CHECK D: WAS THE INVOCA	ATION STATE MACHINE IDLE AT SP A?		

TEST NUMBER: 2.1.4.3.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP B to SP A

Component length: correct number of octets

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y (y is an invalid error code for this operation)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000011 (RETURN ERROR)

Problem code length: 00000001

Problem code: 00000010 (unrecognized error)

TEST	NUMBER: 2.1.4.3.2		Sheet: 1 of 2
REFE	ERENCE: 3.2.2/Q.774		
TITL	E: Valid Functions; Reception of c	component leading to TC-User reject	
SUBT	FITLE: Return Error problem; Une	expected error	
PURF	POSE: To verify that a rejection ca	an be successfully initiated due to an unexpected	d error code included in the Return Error
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component of Class 1 2) Arrange the data at SP B such that a Return Error with an unexpected error code is generated			
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	ECTED MESSAGE AND COMPO	NENT FLOW:	
	SP A (CSL)		SP B (CSL)
	TC-INVOKE req.		
	·		
	INVOKE (i)	<>	RETURN ERROR (i)
	TC-U-ERROR ind.		RETURN ERROR (I)
	TC-U-REJECT req.		
	REJECT (i)	>	
TEST	DESCRIPTION		
1.	Initiate a Class 1 operation invoca Generate an unsuccessful respons operation.	ation from SP A to SP B. se from SP B to SP A with a valid Invoke ID but	an unexpected error code for this
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?		
3.	CHECK B: WAS THE RETUR	N ERROR COMPONENT PASSED TO TC-US	SER BY SP A?
4.	CHECK C: WAS THE REJECT	T COMPONENT SENT BY SP A?	
5.	CHECK D: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?	

TEST NUMBER: 2.1.4.3.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN ERROR)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y (y is an error code that is not one of those which the invoked operation may report)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000011 (RETURN ERROR)

Problem code length: 00000001

Problem code: 00000011 (unexpected error)

TEST	NUMBER: 2.1.4.3.3		Sheet: 1 of 2
REFE	RENCE: 3.2.2/Q.774		
TITL	E: Valid Functions; Reception of c	omponent leading to TC-User reject	
SUBT	TITLE: Return Error problem; Wro	ong type parameter	
PURI	POSE: To verify that a rejection ca	an be successfully initiated due to a wrong type	e parameter included in the Return Error
1) Aı	TEST CONDITIONS: rrange the TC-User stimulus such tass 1	that an appropriate TSL message generated at	SP A contains an Invoke component of
2) A1	rrange the data at SP B such that a R	Leturn Error with a wrong type parameter is gene	rated
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	ECTED MESSAGE AND COMPON	NENT FLOW:	
	SP A (CSL)		SP B (CSL)
	TC-INVOKE req. =====>		
	INVOKE (i)	>	
		<	RETURN ERROR (i)
	TC-U-ERROR ind.		
	<i>TC-U-REJECT req.</i>		
	REJECT (i)	>	
TEST DESCRIPTION			
1.	Initiate a Class 1 operation invoca Generate an unsuccessful respons operation.	ation from SP A to SP B. te from SP B to SP A with a valid Invoke ID but	a wrong type parameter for this
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?
3.	CHECK B: WAS THE RETUR	N ERROR COMPONENT PASSED TO TC-US	SER BY SP A?
4.	CHECK C: WAS THE REJECT	COMPONENT SENT BY SP A?	
5.	CHECK D: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?	

TEST NUMBER: 2.1.4.3.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN ERROR)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y (y is a valid error code for this operation)

parameters (provided by the TC-User, including at least one parameter tag which is not one of those associated with the

outcome of the operation)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000011 (RETURN ERROR)

Problem code length: 00000001

Problem code: 00000100 (wrong type parameter)

TEST	NUMBER: 2.1.5.1.1		Sheet: 1 of 2	
REFE	RENCE: 3.2.1/Q.774			
TITLI	E: Valid Functions; Segmentation	for Return Result		
SUBT	TTLE: Class 1 single operation in	vocation; IUT as sender: receive segmented com	ponents	
PURP	OSE: To verify that a single Class	1 operation can be completed by receiving segm	ented Return Result components	
1) Aı	-	nat an appropriate TSL message generated at SP. Return Result Not-Last component can be genera	-	
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req. ======>			
	INVOKE (i)	>		
	HVVOKE (I)	<	RETURN RESULT NOT-LAST (i)	
	TC-RESULT-NL ind.			
		<	RETURN RESULT-LAST (i)	
	TC-RESULT-L ind. <====================================			
TEST	DESCRIPTION			
1.	Initiate a single operation invocat	cion from SP A to SP B.		
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3.	CHECK B: WAS THE RETURN TO TC-USER BY	RN RESULT NOT-LAST COMPONENT WITH SP A?	I CORRECT INFORMATION PASSED	
4.	CHECK C: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?		
СНЕС	CK TABLE FOR COMPONENTS	WITHIN MESSAGES		
Comp	onent portion in TSL messages			
Co Co	Component portion tag: 01101100 Component portion length: correct number of octets			
INVO	INVOKE component in TSL message from SP A to SP B			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
In	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.5.1.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT NOT-LAST component in TSL message from SP B to SP A

Component type tag: 10100111 (RETURN RESULT NOT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.5.1.2		Sheet: 1 of 2	
REFERENCE: 3.2.1/Q.774			
TITLE: Valid Functions; Segmentation	for Return Result		
SUBTITLE: Class 1 single operation in	vocation; IUT as receiver: send segmented comp	onents	
PURPOSE: To verify that a single Class	s 1 operation can be completed by sending segme	nted Return Result components	
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains an Invoke component 2) Arrange the TC-User stimulus at SP A such that a Return Result Not-Last component can be generated			
CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPO	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE ind.			
	<	INVOKE (i)	
RETURN RESULT NOT-LAST (i)	· >	EVOILE (I)	
TC-RESULT-NL req.			
RETURN RESULT-LAST (i)	>		
TC-RESULT-L req.			
TEST DESCRIPTION			
Initiate a single operation invoca-	tion from SP B to SP A.		
2. CHECK A: WAS THE INVO SP A?	KE COMPONENT WITH CORRECT INFOR	MATION PASSED TO TC-USER BY	
3. CHECK B: WAS THE RETU BY SP A?	RN RESULT NOT-LAST COMPONENT WI	TH CORRECT INFORMATION SENT	
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES		
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct	Component portion tag: 01101100 Component portion length: correct number of octets		
INVOKE component in TSL message from	om SP A to SP B		
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.5.1.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT NOT-LAST component in TSL message from SP B to SP A

Component type tag: 10100111 (RETURN RESULT NOT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

RETURN RESULT LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.5.2.1		Sheet: 1 of 2	
REFERENCE: 3.2.1/Q.774	REFERENCE: 3.2.1/Q.774		
TITLE: Valid functions; Segmentation	for Return Result		
SUBTITLE: Class 3 single operation in	vocation; IUT as sender: Receive segmented con	nponent	
PURPOSE: To verify that a single Class Return Result components	3 operation can be completed by receiving segm	nented	
PRE-TEST CONDITIONS:			
1) Arrange the TC-User stimulus such the	hat an appropriate TSL message generated at SP	A contains an Invoke component	
2) Arrange the data at SP B such that a	Return Result Not-Last component can be genera	ited	
CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPO	I NENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	>		
TC-RESULT-NL ind.	<	RETURN RESULT NOT-LAST (i)	
TC-RESULT-L ind.	<	RETURN RESULT-LAST (i)	
TEST DESCRIPTION			
Initiate a single Class 3 operation	n invocation from SP A to SP B.		
2. CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3. CHECK B: WAS THE RETUIT TO TC-USER BY	RN RESULT NOT-LAST COMPONENT WITH SP A?	H CORRECT INFORMATION PASSED	
4. CHECK C: WAS THE INVOC	CATION STATE MACHINE IDLE AT SP A ?		
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES		
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct	number of octets		
INVOKE component in TSL message from	om SP A to SP B		
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.5.2.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT NOT-LAST component in TSL message from SP B to SP A

Component type tag: 10100111 (RETURN RESULT NOT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST	NUMBER: 2.1.6		Sheet: 1 of 2
REFE	RENCE: 3.2.1/Q.774		
TITLI	E: Valid functions		
SUBT	TTLE: User Cancel		
PURP	OSE: To verify that an operation is	nvocation can be canceled by TC-User	
1) Ar	· ·	nat an appropriate TSL message generated at SP at Return Result-Last component can be generated	A contains an Invoke component
	CONFIGURATION: 1	TYPE OF TEST: VAT and CPT	TYPE OF SP: SP
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:	
	SP A (CSL)		SP B (CSL)
	TC-INVOKE req.		
	INVOKE (i)	>	
	TC-U-CANCEL req. ======>		
		<	RETURN-RESULT-LAST (i)
	TC-L-REJECT ind.		
	<========		
	REJECT (i)	>	
TEST	DESCRIPTION		
1.	Initiate a single Class 1 operation	invocation from SP A to SP R	
1.	Arrange TC-User to cancel the op	peration immediately after the Invoke component	t is sent.
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?
3.	CHECK B: WAS THE REJECT	COMPONENT WITH CORRECT INFORMA	TION SENT BY SP A?
4.	CHECK C: WAS THE COMPO	ONENT FLOW AS SHOWN ABOVE?	
5.	CHECK D: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?	
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Comp	onent portion in TSL messages		
Co	Component portion tag: 01101100 Component portion length: correct number of octets		
INVO	INVOKE component in TSL message from SP A to SP B		
Co Co	Component type tag: 10100001 (INVOKE) Component length: correct number of octets		
In	voke ID tag: 00000010 voke ID length: 00000001 (one oc voke ID: i (i represents an integer)	ctet)	

TEST NUMBER: 2.1.6 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000000 (unrecognized invoke ID)

TEST	NUMBER: 2.1.7.1		Sheet: 1 of 2	
REFE	RENCE: 3.3/Q.773			
TITLE	: Valid functions; Encoding varia	ntions		
SUBTI	ITLE: Component length definite	short		
PURPO	OSE: To verify that a component p	portion with a definite short form can be accepte	d	
1) Arr	PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains an Invoke component 2) Arrange the data at SP A such that a Return Result-Last component can be generated			
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:		
S	SP A (CSL)		SP B (CSL)	
		<	INVOKE (i)	
7	TC-INVOKE ind. <======			
7 =	TC-RESULT-L req.			
_	DETUDN DECLI T I ACT (2)			
F	RETURN RESULT-LAST (i)	>		
TEST	DESCRIPTION			
1.	Initiate a Class 1 or 3 operation in	nvocation from SP B to SP A.		
2.	CHECK A: WAS THE INVO	OKE COMPONENT WITH CORRECT INFO	ORMATION PASSED TO TC-USER	
3.	3. CHECK B: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Compo	Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOI	INVOKE component in TSL message from SP B to SP A			
Co Co	omponent type tag: 10100001 (IN omponent length: correct number of	VOKE) f octets (definite short form)		
Inv	Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.1.7.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP A to SP B

Component type tag: 10100011 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

ID: 1

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.7.2		Sheet: 1 of 2
REFERENCE: 3.3/Q.773		
TITLE: Valid functions; Encoding varia	ations	
SUBTITLE: Component length definite	elong	
PURPOSE: To verify that a component	portion with a definite long form can be accepted	
·	hat an appropriate TSL message generated at SP Return Result-Last component can be generated	B contains an Invoke component
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPECTED MESSAGE AND COMPO	NENT FLOW:	
SP A (CSL)		SP B (CSL)
	<	INVOKE (i)
TC-INVOKE ind.		
TC-RESULT-L req.		
RETURN RESULT-LAST (i)	>	
(4)		
TEST DESCRIPTION		
1. Initiate a Class 1 or 3 operation i	nvocation from SP B to SP A.	
2. CHECK A: WAS THE INVO	OKE COMPONENT WITH CORRECT INFO	ORMATION PASSED TO TC-USER
3. CHECK B: WAS THE RET BY SP A?	URN RESULT-LAST COMPONENT WITH	CORRECT INFORMATION SENT
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES		
Component portion in TSL messages		
Component portion tag: 01101100 Component portion length: correct number of octets		
INVOKE component in TSL message from	om SP B to SP A	
Component type tag: 10100001 (INVOKE) Component length: correct number of octets (definite long)		
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)		

TEST NUMBER: 2.1.7.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00000010 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.7.3		Sheet: 1 of 2
REFERENCE: 3.3/Q.773		
TITLE: Valid functions; Encoding vari	ations	
SUBTITLE: Component length indefin	ite	
PURPOSE: To verify that a component	portion with a indefinite form can be accepted	
	hat an appropriate TSL message generated at SP Return Result-Last component can be generated	B contains an Invoke component
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPECTED MESSAGE AND COMPO	NENT FLOW:	
SP A (CSL)		SP B (CSL)
	<	INVOKE (i)
TC-INVOKE ind.		
TC-RESULT-L req.		
RETURN RESULT-LAST (i)	>	
TEST DESCRIPTION		
1. Initiate a Class 1 or 3 operation i	nvocation from SP B to SP A.	
2. CHECK A: WAS THE INVO	DKE COMPONENT WITH CORRECT INFO	ORMATION PASSED TO TC-USER
3. CHECK B: WAS THE RET BY SP A?	URN RESULT-LAST COMPONENT WITH	CORRECT INFORMATION SENT
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES		
Component portion in TSL messages		
Component portion tag: 01101100 Component portion length: correct number of octets		
INVOKE component in TSL message from SP B to SP A		
Component type tag: 10100001 (INVOKE) Component length: correct number of octets (indefinite form)		
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)		

TEST NUMBER: 2.1.7.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

EOC Tag: 0000 0000 EOC Length: 0000 0000

RETURN RESULT-LAST component in TSL message from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.7.4.1.1	Sheet: 1 of 2		
REFERENCE: 6.2/Q.773			
TITLE: Valid functions; Encoding variations			
SUBTITLE: Value variations; Invoke ID; Invoke ID = -12	7 (FFh)		
PURPOSE: To verify that the IUT (SP A) is able to deal with	th correct encoding of component ID (upper value)		
PRE-TEST CONDITIONS: Arrange the TC-User stimulu Invoke component	s such that an appropriate TSL message generated at SP B contains an		
CONFIGURATION: 1 TYPE	OF TEST: VAT TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPONENT FLOW:			
SP A (CSL)	SP B (CSL)		
<	INVOKE (i)		
TC-INVOKE ind.			
TC-RESULT-L req.			
,			
RETURN RESULT-LAST (i)	·>		
TEST DESCRIPTION			
Initiate a single operation invocation from SP B to SI	P A with Invoke ID set to 11111111.		
2. CHECK A: WAS THE INVOKE COMPONEN BY SP A?			
3. CHECK B: WAS THE RETURN RESULT-LA BY SP A?	AST COMPONENT WITH CORRECT INFORMATION SENT		
4. CHECK C: WAS THE INVOKE ID IN THE RET THE INVOKE COMPONENT?	CHECK C: WAS THE INVOKE ID IN THE RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT?		
	CHECK D: WAS THE OPERATION CODE IN THE RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: 11111111 (FFh)			

TEST NUMBER: 2.1.7.4.1.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: 11111111 (FFh)

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST	TEST NUMBER: 2.1.7.4.1.2		Sheet: 1 of 2	
REFE	ERENCE: 6.2/	Q.773		
TITLI	E: Valid funct	ions; Encoding varia	tions	
SUBT	TITLE: Value	variations; Invoke II	D; Invoke ID = 0 (00h)	
PURP	POSE: To verif	y that the IUT (SP A	a) is able to deal with correct encoding of compo	nent ID (lower value)
PRE-	TEST CONDIT	TIONS: Arrange the Invoke con	ne TC-User stimulus such that an appropriate TS imponent	L message generated at SP B contains an
	CONFIGURA	ATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	ECTED MESSA	AGE AND COMPO	NENT FLOW:	
	SP A (CSL)		SP B (CSL)
			<	INVOKE (i)
	TC-INVOKE in			
	TC-RESULT-L	•		
		=>		
	RETURN RES	SULT-LAST (i)	>	
TEST	DESCRIPTIO	N		
1.	Initiate a single operation invocation from SP B to SP A with Invoke ID set to 0.			
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?			
3.	CHECK B: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
4.	CHECK C: WAS THE INVOKE ID IN THE RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT?			
5.	CHECK D: WAS THE OPERATION CODE IN THE RETURN RESULT-LAST COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP B to SP A				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: 0				

TEST NUMBER: 2.1.7.4.1.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: 0

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.7.4.2		Sheet: 1 of 2	
REFERENCE: 6.3/Q.773			
TITLE: Valid functions; Encoding variations			
SUBTITLE: Value variations; Global operations	on code		
PURPOSE: To verify that a global operation of	code is correctly decoded by TCAP		
PRE-TEST CONDITIONS: Arrange the TC Invoke compor supported operations.	User stimulus such that an appropriate TS nent with a global operation code. The ation	L message generated at SP B contains an global value does not correspond to a	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPONENT	T FLOW:		
SP A (CSL)		SP B (CSL)	
<		INVOKE (i)	
TC-INVOKE ind.			
TC-U-REJECT req.			
======>			
REJECT (i)	>		
TEST DESCRIPTION			
Initiate an operation invocation from S	P B to SP A with a non-supported global of	peration code.	
2. CHECK A: WAS THE REJECT COL	MPONENT WITH CORRECT INFORMA	TION SENT BY SP A?	
3. CHECK B: WAS THE INVOKE ID IN THE REJECT COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT?			
CHECK TABLE FOR COMPONENTS WITH	IIN MESSAGES		
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			
Operation code tag: 00000110 (global) Operation code length: 00000011 (3) Operation code: 0000 0000 0001 0001 1000 0101			

TEST NUMBER: 2.1.7.4.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

REJECT component in TSL message from SP A to SP B

Component type tag: 10100001 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000001 (INVOKE problem type) Problem code length: 00000001 Problem code: 00000001 (unrecognized operation)

TEST	TEST NUMBER: 2.1.8.1		Sheet: 1 of 2	
REFE	ERENCE: Q.774			
TITL	E: Valid functions; Multiple comp	onents grouping		
SUBT	FITLE: Multiple operations invoca	ation; receiving success		
PURI	POSE: To verify that multiple oper be received	rations can be successfully invoked and the succ	cessful completions of the operations can	
1) A	PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains multiple components 2) Arrange the TC-User at SP B to send successful completions with an appropriate TSL message			
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
	ECTED MESSAGE AND COMPOSE SP A (CSL) TC-INVOKE req. (#1)	INVOKE #1,, #n ^{a)}	SP B (CSL)	
		RETURN RESULT-LAST #1,, #n ^a)		
TC-RESULT-L ind. (#1)				
TEST DESCRIPTION				
1.	Initiate multiple operations within a TSL message from SP A to SP B.			
2.	2. CHECK A: WERE ALL THE INVOKE COMPONENTS WITHIN A TSL MESSAGE SENT BY SP A WITH CORRECT INFORMATION?			
3.	3. CHECK B: WERE ALL THE RETURN-LAST COMPONENTS INSIDE A TSL MESSAGE PASSED TO TC-USER IN THE SAME ORDER AS PROVIDED BY SP B WITH CORRECT INFORMATION?			
4.	CHECK C: WERE ALL THE INVOKE STATE MACHINES (1,, n) IDLE AT SP A?			

TEST NUMBER: 2.1.8.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)

Invoke ID: 1, or, ..., n corresponding to the INVOKE #1, ..., #n

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x1, ..., xn representing valid operation codes

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: 1, or, ..., n

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x1, or, ..., xn (see Note)

parameters (provided by the TC-User)

TEST NUMBER: 2.1.8.2		Sheet: 1 of 2	
REFE	REFERENCE: Q.774		
TITL	E: Valid functions; Multiple	components grouping	
SUBT	TITLE: Multiple operations	invocation; reporting success	
PURI	POSE: To verify that multiple be sent	le operations can be successfully invoked and the succ	cessful completions of the operations can
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains multiple components 2) Arrange the TC-User at SP A to send successful completions with an appropriate TSL message			
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
	CCTED MESSAGE AND CO	OMPONENT FLOW: INVOKE #1,, #n ^a)	SP B (CSL)
TC-INVOKE ind. (#n) TC-RESULT-L req. (#1) TC-RESULT-L req. (#n) RETURN RESULT-LAST #n,, #1a) The sequence of the components is provided by the TC-User NOTE – Number of components is subject to the TC-User			
TEST DESCRIPTION			
 Initiate multiple operations within a TSL message from SP B to SP A. CHECK A: WERE ALL THE INVOKE COMPONENTS WITHIN A TSL MESSAGE PASSED TO TC-USER IN THE SAME ORDER AS PROVIDED BY SP B WITH CORRECT INFORMATION? 			
3.	3. CHECK B: WERE ALL THE RETURN RESULT-LAST COMPONENTS WITHIN A TSL MESSAGE SENT BY SP A WITH CORRECT INFORMATION?		
4.	4. CHECK C: WAS THE INVOKE ID IN EACH OF THE RETURN RESULT-LAST COMPONENTS ONE-TO-ONE CORRESPONDENT WITH THE ONE IN EACH OF THE INVOKE COMPONENTS?		

TEST NUMBER: 2.1.8.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP B to SP A

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)

Invoke ID: 1, or, ..., n corresponding to the INVOKE #1, ..., #n

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x1, ..., xn representing valid operation codes

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: 1, or, ..., n

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note) Operation code: x1, or, ..., xn (see Note)

operation code. A1, o1, ..., All (see 1 total

parameters (provided by the TC-User)

TEST NUMBER: 2.1.8.3 Sheet: 1 of 3

REFERENCE: 3.2.2.2/Q.774

TITLE: Valid functions; Multiple components grouping

SUBTITLE: A malformed component received

PURPOSE: To verify that subsequent components in the message can be discarded when a badly structured component is

detected by the component sublayer

PRE-TEST CONDITIONS: Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains

multiple components, the second of which is badly structured

CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP

EXPECTED MESSAGE AND COMPONENT FLOW:

SP A (CSL) SP B (CSL)

INVOKE #1, #2, #3 (Note 1)
(#2 badly structured, e.g.

operation code missing)

TC-INVOKE ind. (#1)

TC-L-REJECT ind. (#2)

TC-RESULT-L req. (#1)

REJECT #2, RETURN RESULT-LAST #1

(Note 2)

NOTES

- 1 The sequence of the Invoke components is important
- 2 The sequence of these components is not important

TEST DESCRIPTION

- 1. Initiate multiple operations within a TSL message from SP B to SP A with the order shown in the diagram.
- 2. CHECK A: WAS THE FIRST INVOKE COMPONENT PASSED TO TC-USER?
- 3. CHECK B: WERE ONLY THE RETURN RESULT-LAST COMPONENT FOR THE FIRST OPERATION AND THE REJECT COMPONENT FOR THE SECOND OPERATION SENT BY SP A?

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

TEST NUMBER: 2.1.8.3 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

INVOKE #1 component in TSL message from SP B to SP A

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)

Invoke ID: 1

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x is a valid operation code

parameters (provided by the TC-User)

INVOKE #2 component in TSL message from SP B to SP A

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)

Invoke ID: 2

parameters (provided by the TC-User)

INVOKE #3 component in TSL message from SP B to SP A

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)

Invoke ID: 3

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x is a valid operation code

parameters (provided by the TC-User)

TEST NUMBER: 2.1.8.3 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

RETURN RESULT-LAST #1 component in TSL message from SP A to SP B

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: 1

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x

parameters (provided by the TC-User)

REJECT #2 component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: 2

Problem code tag: 10000000 (General Problem)

Problem code length: 00000001
Problem code: 00000010 (badly structured component)

NOTE - Omitted when no parameter is present

TEST NUMBER: 2.2.1.1		Sheet: 1 of 2	
REFERENCE: 6.2/Q.773			
TITLE: Syntactically invalid behavior; In	nvalid values for information elements		
SUBTITLE: Length of Invoke ID >1 in I	nvoke component		
PURPOSE: To verify that a rejection of (value out of range)	a requested operation can be performed due	e to incorrect encoding of component ID	
PRE-TEST CONDITIONS: Arrange the component	e stimulus such that an appropriate TSL messa with an error as described below	age generated at SP B contains an Invoke	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPON	ENT FLOW:		
SP A (CSL)		SP B (CSL)	
	<	INVOKE (i)	
TC-L-REJECT ind.			
<=========			
REJECT (NULL)	>		
TEST DESCRIPTION			
Initiate an operation invocation from the last of the last operation invocation from the last operation from the last ope	om SP B to SP A with Invoke ID equal to 2 oct	ets (illegal value).	
2. CHECK A: WAS THE REJECT	2. CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000010 (two oct Invoke ID: 129	tets)		

TEST NUMBER: 2.2.1.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type)

Problem code length: 00000001

Problem code: 00000001 (mistyped component)

TEST	NUMBER: 2.2.1.2		Sheet: 1 of 2
REFE	REFERENCE: 6.2/Q.773		
TITL	E: Syntactically invalid behavior; Ir	avalid values for information elements	
SUBT	TTLE: Length of Invoke ID = 0 in	Invoke component	
PURF	OSE: To verify that a rejection of (length equals 0)	a requested operation can be performed due	to incorrect encoding of component ID
PRE-	TEST CONDITIONS: Arrange the component	e stimulus such that an appropriate TSL messa with an error as described below	ge generated at SP B contains an Invoke
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE AND COMPON	ENT FLOW:	
	SP A (CSL)		SP B (CSL)
		<	INVOKE
	TC-L-REJECT ind.		
	DE IECT AIII I		
	REJECT (NULL)	·····>	
TEST	DESCRIPTION		
1.	Initiate an operation invocation fro	om SP B to SP A with Invoke ID equal to 0 octo	ets (illegal value).
2.	CHECK A: WAS THE REJECT	COMPONENT WITH CORRECT INFORMA	TION SENT BY SP A?
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVC	INVOKE component in TSL message from SP B to SP A		
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
In In	voke ID tag: 00000010 voke ID length: 00000000 (zero oc	tet)	

TEST NUMBER: 2.2.1.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type)

Problem code length: 00000001

Problem code: 00000001 (wrong type component)

TEST NUMBER: 2.2.2.1.1	Sheet: 1 of 2		
REFERENCE: 6.2/Q.773			
TITLE: Syntactically invalid behavior; Invalid structure			
SUBTITLE: Invoke component; Invoke ID missing			
PURPOSE: To verify that a rejection of a requested operation can be performed due	to Invoke ID missing		
PRE-TEST CONDITIONS: Arrange the stimulus such that an appropriate TSL n component with an error as described below	nessage generated at SP B contains an Invoke		
CONFIGURATION: 1 TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPONENT FLOW:			
SP A (CSL)	SP B (CSL)		
<<	INVOKE		
TC I DE IECT :I			
TC-L-REJECT ind. <==========			
REJECT (NULL)	>		
TEST DESCRIPTION			
1EST DESCRIPTION			
Initiate a single operation invocation from SP B to SP A with Invoke ID miss	sing.		
2. CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFO	RMATION SENT BY SP A?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Operation code tag: 00000110 (global) Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) Operation code: x (x represents a valid operation code)			
parameters (provided by the TC-User)			

TEST NUMBER: 2.2.2.1.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type) Problem code length: 00000001 Problem code: 00000001 (wrong type component)

TEST NUMBER: 2.2.2.1.2	Sheet: 1 of 2		
REFERENCE: 3.2.2.2/Q.774			
TITLE: Syntactically invalid behavior; Invalid structure			
SUBTITLE: Invoke component; Operation code missing			
PURPOSE: To verify that a rejection of a requested operation can be performed due to operation	peration code missing		
PRE-TEST CONDITIONS: Arrange the stimulus such that an appropriate TSL message component with a syntax error as described below	ge generated at SP B contains an Invoke		
CONFIGURATION: 1 TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPONENT FLOW:			
SP A (CSL)	SP B (CSL)		
<	INVOKE (i)		
TC I DE IECT : I	,		
TC-L-REJECT ind. <==========			
REJECT (i)>			
TEST DESCRIPTION			
1. Initiate an operation invocation from SP B to SP A with operation code missing.			
2. CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION	TION SENT BY SP A?		
3. CHECK B: WAS THE INVOKE ID IN THE REJECT COMPONENT THE S COMPONENT?	AME AS THE ONE IN THE INVOKE		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			
parameters (provided by the TC-User)			

TEST NUMBER: 2.2.2.1.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000000 (General problem type) Problem code length: 00000001 Problem code: 00000001 (wrong type component)

TEST NUMBER: 2.2.2.2.1		Sheet: 1 of 2	
REFERENCE: 3.2.2/Q.774			
TITLE: Syntactically invalid behavior; I	invalid structure		
SUBTITLE: Return Result component;	Invoke ID missing		
PURPOSE: To verify that a rejection ca component	n be successfully initiated due to the absence of	f the Invoke ID in the Return Result-Last	
PRE-TEST CONDITIONS:			
1) Arrange the TC-User stimulus such th	at an appropriate TSL message generated at SP.	A contains an Invoke component	
2) Arrange the data at SP B such that a R	Return Result-Last without an Invoke ID is gener	ated	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPOR	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	>		
	<	RETURN RESULT-LAST	
TC-L-REJECT ind.			
REJECT (NULL) time expiry for invocation (i)	>		
TEST DESCRIPTION			
1. Initiate a Class 1 or 3 operation in Generate a response from SP B to	nvocation from SP A to SP B. O SP A without an Invoke ID.		
2. CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3. CHECK B: WAS THE REJECT	3. CHECK B: WAS THE REJECT COMPONENT SENT BY SP A?		
4. CHECK C: WAS THE INVOC	4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP A to SP B			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.2.2.2.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: y (y is different from x) (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type)

Problem code length: 00000001

Problem code: 00000001 (wrong type component)

NOTE - Omitted when no parameter is present

TEST NUMBER: 2.2.2.2.2	TEST NUMBER: 2.2.2.2.2 Sheet: 1 of 2		
REFERENCE: 3.2.2/Q.774	REFERENCE: 3.2.2/Q.774		
TITLE: Syntactically invalid behavior; I	nvalid structure		
SUBTITLE: Return Result component;	Operation code missing while parmeters include	d	
PURPOSE: To verify that a rejection Result-Last component	can be successfully initiated due to the opera	ation code being missing in the Return	
Class 1 or 3	that an appropriate TSL message generated at seturn Result-Last without an operation code is g	-	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPON	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	>		
• •	<	RETURN RESULT-LAST (i)	
TC-L-REJECT ind.			
<=====================================	>		
REJECT (i)			
TEST DESCRIPTION			
Initiate an operation invocation fr Generate a response from SP B to	om SP A to SP B. SP A with a valid Invoke ID but a different ope	eration code.	
2. CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3. CHECK B: WAS THE REJECT	COMPONENT SENT BY SP A?		
4. CHECK C: WAS THE INVOC.	ATION STATE MACHINE IDLE AT SP A?		
CHECK TABLE FOR COMPONENTS V	WITHIN MESSAGES		
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct n	number of octets		
INVOKE component in TSL message fro	m SP A to SP B		
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one oc Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.2.2.2.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000

Sequence length: correct number of octets

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000000 (General problem type)

Problem code length: 00000001

Problem code: 00000001 (wrong type component)

TEST NUMBER: 2.2.2.2.3	EST NUMBER: 2.2.2.2.3 Sheet: 1 of 2		
REFERENCE: 6.4/Q.773; 3.2.2.2/Q.774			
TITLE: Syntactically invalid behavior	; Invalid structure		
SUBTITLE: Return Result componer	t; Sequence tag missing while parmeters included		
PURPOSE: To verify that a rejection of	can be successfully initiated due to Sequence tag m	issing while parameters included	
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that an appropriate TSL message contains a Return Result-Last component with an invalid Sequence tag			
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMP	ONENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req. ====>			
INVOKE (i)	>		
	<	RETURN RESULT-LAST (i)	
TC-L-REJECT ind. <======			
REJECT (i)	>		
TEST DESCRIPTION			
Initiate an operation invocation	from SP A to SP B.		
2. CHECK A: WAS THE INVO	KE COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3. CHECK B: WAS THE REJE	CT COMPONENT WITH CORRECT INFORMA	TION SENT BY SP A?	
4. CHECK C: WAS THE INVO	4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP A to SP B			
Component type tag: 10100001 (I Component length: correct numbe	NVOKE) of octets		
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one Invoke ID: i (i represents an integ			

TEST NUMBER: 2.2.2.2.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000000 (General problem type)

Problem code length: 00000001

Problem code: 00000001 (wrong type component)

TEST NUMBER: 2.2.2.3.1		Sheet: 1 of 2	
REFERENCE: 3.2.2/Q.774	REFERENCE: 3.2.2/Q.774		
TITLE: Syntactically invalid behavior; In	valid structure		
SUBTITLE: Return Error; Invoke ID mis	sing		
PURPOSE: To verify that a rejection can Invoke ID in the Return Error	be successfully initiated due to the absence of to component	he	
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such the Class 1	at an appropriate TSL message generated at SF	A contains an Invoke component of the	
	eturn Error without an Invoke ID is generated		
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPON	ENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	>		
	<	RETURN ERROR	
TC-L-REJECT ind.			
REJECT (NULL) time expiry for invocation (i)	>		
TEST DESCRIPTION			
Initiate a Class 1 operation invocat Generate an unsuccessful response	ion from SP A to SP B. from SP B to SP A without an Invoke ID.		
2. CHECK A: WAS THE INVOKE	COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3. CHECK B: WAS THE REJECT	COMPONENT SENT BY SP A?		
4. CHECK C: WAS THE INVOCA	TION STATE MACHINE IDLE AT SP A?		
CHECK TABLE FOR COMPONENTS W	/ITHIN MESSAGES		
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct nu	umber of octets		
INVOKE component in TSL message from	n SP A to SP B		
Component type tag: 10100001 (INV Component length: correct number of			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one oct Invoke ID: i (i represents an integer)	et)		

TEST NUMBER: 2.2.2.3.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN ERROR)

Component length: correct number of octets

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long) Error code: y (y is an error code which the invoked operation may report)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type)

Problem code length: 00000001

Problem code: 00000001 (mistyped component)

TEST	NUMBER: 2.2.2.3.2		Short, 1 of 2	
			Sheet: 1 of 2	
REFE	RENCE: 3.2.2/Q.774			
TITLI	E: Syntactically Invalid Behavior;	Invalid structure		
SUBT	ITLE: Return Error; Error code m	issing		
PURP	OSE: To verify that a rejection component	can be successfully initiated due to the absence	e of the error code in the Return Error	
PRE-	TEST CONDITIONS:			
	range the TC-User stimulus such ass 1	that an appropriate TSL message generated at	SP A contains an Invoke component of	
2) A1	range the data at SP B such that a R	Return Error without an error code is generated		
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPOR	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	>		
		<	RETURN ERROR (i)	
TC-L-REJECT ind.				
	<=====================================			
REJECT (i)				
TEST	DESCRIPTION			
1.	Initiate a Class 1 operation invoca Generate an unsuccessful respons	ation from SP A to SP B. se from SP B to SP A with a valid Invoke ID but	without error code for this operation	
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3.	CHECK B: WAS THE REJECT	Γ COMPONENT SENT BY SP A?		
4.	CHECK C: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Comp	onent portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP A to SP B				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
In	voke ID tag: 00000010 voke ID length: 00000001 (one or	otet)		

TEST NUMBER: 2.2.2.3.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN ERROR) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000101 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000000 (General problem type)

Problem code length: 00000001

Problem code: 00000001 (mistyped component)

TEST NUMBER: 2.2.2.4.1 Sheet: 1 of 1

REFERENCE: 3.2.2.2/Q.774

TITLE: Syntactically Invalid Behavior; Invalid structure

SUBTITLE: Unknown component type; Invoke ID unrecognizable

PURPOSE: To verify that a rejection can be initiated due to Unknown component type with unrecognized Invoke ID

PRE-TEST CONDITIONS: Arrange the stimulus such that an appropriate TSL message generated at SP B contains an Unknown

component as described below

CONFIGURATION: 1 TYPE OF TEST: VAT TYPE OF SP: SP

EXPECTED MESSAGE AND COMPONENT FLOW:

SP A (CSL) SP B (CSL)

<----- Unknown component

TC-L-REJECT ind.

REJECT (NULL)

TEST DESCRIPTION

1. Initiate an operation invocation from SP B to SP A with an Unknown component type with any content.

2. CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

Unknown component in TSL message from SP B to SP A

Component type tag: any values except 10100001, 10100010, 10100011, 10100100 and 10100111

Component length: correct number of octets

Component content: any

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type)

Problem code length: 00000001

Problem code: 00000000 (unrecognized component)

TEST NUMBER: 2.2.2.4.2		Sheet: 1 of 2	
REFERENCE: 3.2.2.2/Q.774			
TITLE: Syntactically Invalid Behavior;	Invalid structure		
SUBTITLE: Unknown component type	; Invoke ID derivable		
PURPOSE: To verify that a rejection ca	nn be initiated due to Unknown component type v	with derivable Invoke ID	
PRE-TEST CONDITIONS: Arrange the component	e stimulus such that an appropriate TSL message with a derivable Invoke ID as described below	generated at SP B contains an Unknown	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPO	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
	<	Unknown component (i)	
TC-L-REJECT ind.			
REJECT (i or NULL)	>		
TEST DESCRIPTION			
1. Initiate an operation invocation from SP B to SP A with an Unknown component type as described below.			
2. CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES		
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
Unknown component in TSL message from SP B to SP A			
Component type tag: any values except 10100001, 10100010, 10100011, 10100100 and 10100111 Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			
Operation code tag: 00000010 (local) or 00000110 (global) Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) Operation code: x (x represents an operation code)			
parameters (provided by the TC-User)			

TEST NUMBER: 2.2.2.4.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: i

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type) Problem code length: 00000001 Problem code: 00000000 (unrecognized component)

TEST NUMBER: 2.2.3.1	TEST NUMBER: 2.2.3.1		
REFERENCE: 3.2.2.2/Q.774; 3.2.3/Q.7	REFERENCE: 3.2.2.2/Q.774; 3.2.3/Q.773		
TITLE: Syntactically Invalid Behavior;	Invalid encoding for Invoke component		
SUBTITLE: Invalid tag			
PURPOSE: To verify that a rejection is	generated because of an invalid tag		
PRE-TEST CONDITIONS: Arrange the component	e stimulus such that an appropriate TSL messag with an error as described below	ge generated at SP B contains an Invoke	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPO	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
	<	INVOKE (i)	
TC-L-REJECT ind.			
REJECT (i or NULL)	>		
REGET (FOI NEEL)			
TEST DESCRIPTION			
1. Initiate an operation invocation from SP B to SP A with an invalid tag.			
2. CHECK A: WAS THE REJECT	2. CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?		
CHECK TABLE FOR COMPONENTS	WITHIN MESSAGES		
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.2.3.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Invalid tag: 00011111

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type)
Problem code length: 00000001
Problem code: 00000010 (badly structured component)

TEST	TEST NUMBER: 2.2.3.2		Sheet: 1 of 2
REFE	ERENCE: 3.2.2.2/Q.774		
TITL	E: Syntactically Invalid Behavior;	Invalid encoding for Invoke component	
SUBT	FITLE: Wrong component length		
PURI	POSE: To verify that a rejection of	a requested operation can be initiated due to wro	ong component length
PRE-	TEST CONDITIONS: Arrange the component	e stimulus such that an appropriate TSL messag with a syntax error as described below	ge generated at SP B contains an Invoke
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPI	ECTED MESSAGE AND COMPO	NENT FLOW:	
	SP A (CSL)		SP B (CSL)
		<	INVOKE (i)
	<i>TC-L-REJECT ind.</i> <====================================		
	REJECT (i or NULL)	>	
	,		
TEST DESCRIPTION			
1.	1. Initiate an operation invocation from SP B to SP A with an invalid component length value.		
2.	CHECK A: WAS THE REJECT	Γ COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?
3.	CHECK B: WAS THE INVOK COMPONENT?	KE ID IN THE REJECT COMPONENT THE S	SAME AS THE ONE IN THE INVOKE
CHE	CK TABLE FOR COMPONENTS	WITHIN MESSAGES	
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP B to SP A			
Component type tag: 10100001 (INVOKE) Component length: wrong number of octets (e.g. 00000000)			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.2.3.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)
Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: i

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type)
Problem code length: 00000001
Problem code: 00000010 (badly structured component)

TEST NUMBER: 2.2.3.3		Sheet: 1 of 2		
REFERENCE: 3.3/Q.773				
TITLE: Syntactically Invalid Behavior;	Invalid encoding for Invoke component			
SUBTITLE: Missing EOC in indefinite	form			
PURPOSE: To verify that a component	portion with an indefinite form but EOC missing	g is rejected		
PRE-TEST CONDITIONS: Arrange the stimulus such that an appropriate TSL message generated at SP B contains an Invoke component				
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPON	NENT FLOW:			
SP A (CSL)		SP B (CSL)		
	<	INVOKE (i)		
TC-L-REJECT ind.				
REJECT (i or NULL)	>			
TEST DESCRIPTION				
Initiate a single operation invocat	ion from SP B to SP A.			
2. CHECK A: WAS THE REJECT	COMPONENT WITH CORRECT INFORMA	TION SENT BY SP A?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP B to SP A				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets (indefinite form)				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.2.3.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: i

NULL tag: 00000101 NULL length: 00000000

Problem code tag: 10000000 (General problem type)
Problem code length: 00000001
Problem code: 00000010 (badly structured component)

TEST NUMBER: 2.3.1.1		Sheet: 1 of 2		
REFERENCE: 3.2.2/Q.774				
TITLE: Inopportune Behavior; Inopportune Invoke component				
SUBTITLE: Invalid linked ID				
PURPOSE: To verify that a rejection of a	a requested operation can be initiated due to inv	alid linked ID		
PRE-TEST CONDITIONS: 1) Arrange the stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that a linked Invoke component can be generated as described below				
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPON SP A (CSL) TC-INVOKE req. INVOKE (i)	IENT FLOW:	SP B (CSL)		
TC-L-REJECT ind. <===================================	<>	INVOKE (j, k)		
TEST DESCRIPTION				
Initiate an operation invocation from	om SP A to SP B.			
2. CHECK A: WAS THE INVOKE	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?		
3. CHECK B: WAS THE REJECT	. CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
	CHECK C: WAS THE INVOKE ID IN THE REJECT COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT SENT BY SP B?			
5. CHECK D: WAS THE INVOCA	ATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages Component portion tag: 01101100 Component portion length: correct number of octets INVOKE component in TSL message from SP A to SP B Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet)				
Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.3.1.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

INVOKE component in TSL message sent by SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet)
Invoke ID: j (j represents an integer)

Linked ID tag: 10000000

Linked ID length: 00000001 (one octet)

Linked ID: k (k is an integer which is different from i)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)

Operation code: y (y represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in the TSL message sent by SP A

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001

Problem code: 00000101 (unrecognized linked ID)

TEST	NUMBER: 2.3.2.1		Sheet: 1 of 2	
REFERENCE: 3.2.2/Q.774				
TITLE: Inopportune Behavior; Unrecognized Invoke ID				
SUBT	TITLE: Inopportune Return Result	-Last component		
PURPOSE: To verify that a rejection can be successfully initiated due to an unrecognized Invoke ID (never used and just released) in the received Return Result-Last component				
PRE-	TEST CONDITIONS:			
 Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component for operation Class 1 or 3 Arrange the data at SP B such that a Return Result-Last with an invalid Invoke ID is generated 				
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CCTED MESSAGE AND COMPO	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	>		
		<	RETURN RESULT-LAST (j)	
	TC-L-REJECT ind.			
	REJECT (j) time expiry for invocation (i)	>		
	TC-L-CANCEL ind.			
		<	RETURN RESULT-LAST (i)	
	TC-L-REJECT ind.			
	REJECT (i)	>		
TEST DESCRIPTION				
1.	Initiate an operation invocation fr Generate a response from SP B to	om SP A to SP B. SP A with an unrecognized Invoke ID.		
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3.	CHECK B: WAS THE REJECT COMPONENT SENT BY SP A ?			
4.	Generate a Return Result-Last co	mponent from SP B to SP A.		
5.	CHECK C: WAS THE REJECT	COMPONENT SENT BY SP A?		
6	CHECK D. WAS THE COMPO	ONENT ELOW AS SHOWN IN ABOVE 2		

TEST NUMBER: 2.3.2.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: j (j is different from i)

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000000 (unrecognized invoke ID)

The contents of the last two components, RETURN-RESULT-LAST (i) and REJECT (i), are the same as above except the

Invoke ID is (i)

NOTE - Omitted when no parameter is present.

TEST	NUMBER: 2.3.2.2		Sheet: 1 of 3	
REFERENCE: 3.2.2/Q.774				
TITLE: Inopportune Behavior; Unrecognized Invoke ID				
SUBT	TITLE: Inopportune Return Result	Not-Last component		
PURPOSE: To verify that a rejection can be successfully initiated due to an unrecognized Invoke ID (never used and just released) in the received Return Result Not-Last component				
PRE-	TEST CONDITIONS:			
 Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component for operation Class 1 or 3 Arrange the data at SP B such that a Return Result Not-Last with an invalid Invoke ID is generated 				
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPOR	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	>		
		<	RETURN RESULT NOT-LAST (j)	
	TC-L-REJECT ind.			
	REJECT (j) time expiry for invocation (i)	>		
	TC-L-CANCEL ind.			
		<	RETURN RESULT NOT-LAST (i)	
	TC-L-REJECT ind.			
	REJECT (i)	>		
TEST DESCRIPTION				
1.	Initiate an operation invocation fr Generate a response from SP B to	om SP A to SP B. SP A with an unrecognized Invoke ID.		
2.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3.	CHECK B: WAS THE REJECT COMPONENT SENT BY SP A ?			
4.	Generate a Return Result Not-Las	st component from SP B to SP A.		
5.	CHECK C: WAS THE REJECT	COMPONENT SENT BY SP A?		
6	CHECK D: WAS THE COMPO	ONENT ELOW AS SHOWN IN ABOVE 2		

TEST NUMBER: 2.3.2.2 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT NOT-LAST component in TSL message from SP B to SP A

Component type tag: 10100111 (RETURN RESULT NOT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: j (j is different from i)

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

TEST NUMBER: 2.3.2.2 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000000 (unrecognized invoke ID)

RETURN RESULT NOT-LAST component in TSL message from SP B to SP A

Component type tag: 10100111 (RETURN RESULT NOT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000000 (unrecognized invoke ID)

TEST	NUMBER: 2.3.2.3		Sheet: 1 of 3
REFE	RENCE: 3.2.2/Q.774		
TITLI	E: Inopportune Behavior; Unrecog	nized Invoke ID	
SUBT	TITLE: Inopportune Return Error o	component	
PURP	POSE: To verify that a rejection of released) in the received Ret	can be successfully initiated due to an unreco urn Error component	gnized Invoke ID (never used and just
PRE-	TEST CONDITIONS:		
un	recognized operation Class 1 or 2	that an appropriate TSL message generated at a	
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CCTED MESSAGE AND COMPO	NENT FLOW:	
	SP A (CSL)		SP B (CSL)
	TC-INVOKE req.		
	INVOKE (i)	>	
		<	RETURN ERROR (j)
	TC-L-REJECT ind.		
	REJECT (j) time expiry for invocation (i)	>	
	TC-L-CANCEL ind.		
		<	RETURN ERROR (i)
	TC-L-REJECT ind.		
	REJECT (i)	>	
TEST DESCRIPTION			
1.	Initiate an operation invocation fr Generate an unsuccessful respons	om SP A to SP B. e from SP B to SP A with an invalid Invoke ID.	
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?
3.	CHECK B: WAS THE REJECT	COMPONENT SENT BY SP A ?	
4.	Generate a Return Error compone	ent from SP B to SP A.	
5.	CHECK C: WAS THE REJECT	COMPONENT SENT BY SP A?	
6	CHECK D. WAS THE COMPO	NENT ELOWAS ABOVE 2	

TEST NUMBER: 2.3.2.3 Sheet: 2 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN ERROR) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: j (j is different from i)

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: j

Problem code tag: 10000011 (RETURN ERROR)

Problem code length: 00000001

Problem code: 00000000 (unrecognized invoke ID)

TEST NUMBER: 2.3.2.3 Sheet: 3 of 3

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

RETURN ERROR component in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN ERROR) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000011 (RETURN ERROR) Problem code length: 00000001

Problem code: 00000000 (unrecognized invoke ID)

TEST	NUMBER: 2.3.2.4		Sheet: 1 of 2	
REFE	RENCE: 3.2.2/Q.774			
TITLE	E: Inopportune Behavior; Unrecog	gnized Invoke ID		
SUBT	TTLE: Inopportune Reject compo	nent		
PURP	OSE: To verify that receipt of a effect on an active invocation	Reject component with an Invoke ID not corres	sponding to any active invocation has no	
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component for Class 1 or 2 2) Arrange the data at SP B such that a Reject with an unrecognized Invoke ID is generated				
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:		
	SP A (CSL)		SP B (CSL)	
	TC-INVOKE req.			
	INVOKE (i)	>		
	TC-R-REJECT ind.a)	<	REJECT (j)	
	TC-L-RESULT ind.	<	RETURN RESULT-LAST (i)	
a) The issuing of the TC-R-REJECT ind. is implementation dependent.				
TEST DESCRIPTION				
1.	Initiate an operation invocation f Generate a reject from SP B to S.			
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?	
3.	Generate a Reject component fro	m SP B to SP A.		
4.	CHECK B: WAS THE COMPO	ONENT FLOW AS ABOVE?		
5.	CHECK C: WAS THE INVOC	ATION STATE MACHINE IDLE AT SP A?		

TEST NUMBER: 2.3.2.4 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Component portion in TSL messages

Component portion tag: 01101100

Component portion length: correct number of octets

INVOKE component in TSL message from SP A to SP B

Component type tag: 10100001 (INVOKE) Component length: correct number of octets

Invoke ID tag: 00000010

Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001 Invoke ID: j (j is different from i)

Problem code tag: 10000001 (INVOKE)

Problem code length: 00000001 Problem code: any value

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000101 (Global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

TEST NUMBER: 2.3.3.1		Sheet: 1 of 2	
REFERENCE: 3.2.1/Q.774			
TITLE: Inopportune Behavior; Unexpe	cted Components		
SUBTITLE: Return Result-Last for Cla	uss 2		
PURPOSE: To verify that a rejection ca	an be sent if a Return Result-Last component is re	eceived for a Class 2 operation	
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that a Return Result-Last component can be generated			
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPO	NENT FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	>		
TC-L-REJECT ind. <====================================	>	RETURN RESULT-LAST (i)	
TEST DESCRIPTION			
1. Initiate a Class 2 operation invoc	ation from SP A to SP B.		
	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?		
4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A? CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP A to SP B			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.3.3.1 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000001 (return result unexpected)

TEST NUMBER: 2.3.3.2		Sheet: 1 of 2		
REFERENCE: 3.2.1/Q.774				
TITLE: Inopportune Behavior; Unexp	ected Components			
SUBTITLE: Return Result-Last for Cl	ass 4			
PURPOSE: To verify that a rejection of	an be sent if a Return Result-Last component is re	eceived for a Class 4 operation		
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that a Return Result-Last component can be generated				
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPO	ONENT FLOW:			
SP A (CSL) TC-INVOKE req. TNVOKE (i)	>	SP B (CSL)		
INVOKE (i)	<>	RETURN RESULT-LAST (i)		
TC-L-REJECT ind.				
REJECT (i)	>			
TEST DESCRIPTION				
Initiate a Class 4 operation invo	cation from SP A to SP B.			
2. CHECK A: WAS THE INVO	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3. CHECK B: WAS THE REJEC	CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
4. CHECK C: WAS THE INVO	4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP A to SP B				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.3.3.2 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT-LAST component in TSL message from SP B to SP A

Component type tag: 10100010 (RETURN RESULT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000001 (return result unexpected)

TEST NUMBER: 2.3.3.3		Sheet: 1 of 2		
REFERENCE: 3.2.1/Q.774				
TITLE: Inopportune Behavi	or; Unexpe	ected Components		
SUBTITLE: Return Result N	lot-Last fo	r Class 2		
PURPOSE: To verify that a	rejection ca	an be sent if a Return Result Not-Last component	t is received for a Class 2 operation	
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that a Return Result Not-Last component can be generated				
CONFIGURATION:	1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AN	О СОМРО	NENT FLOW:		
SP A (CSL) TC-INVOKE req. TOWN (2)			SP B (CSL)	
INVOKE (i)		<>	RETURN RESULT NOT-LAST (i)	
TC-L-REJECT ind.				
<=====================================		>		
RESECT (I)				
TEST DESCRIPTION				
1. Initiate a Class 2 oper	1. Initiate a Class 2 operation invocation from SP A to SP B.			
2. CHECK A: WAS T	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3. CHECK B: WAS T	CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
4. CHECK C: WAS T	4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP A to SP B				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.3.3.3 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT NOT-LAST component in TSL message from SP B to SP A

Component type tag: 10100111 (RETURN RESULT NOT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000001 (return result unexpected)

TEST NUMBER: 2.3.3.4		Sheet: 1 of 2		
REFERENCE: 3.2.1/Q.774				
TITLE: Inopportune Behavior; Unexpe	cted Components			
SUBTITLE: Return Result Not-Last for	r Class 4			
PURPOSE: To verify that a rejection ca	an be sent if a Return Result Not-Last component	is received for a Class 4 operation		
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component 2) Arrange the data at SP B such that a Return Result Not-Last component can be generated				
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP		
EXPECTED MESSAGE AND COMPO	NENT FLOW:			
SP A (CSL)		SP B (CSL)		
TC-INVOKE req.				
INVOKE (i)	>			
	<	RETURN RESULT NOT-LAST (i)		
TC-L-REJECT ind. <======				
REJECT (i)				
TEST DESCRIPTION				
Initiate a Class 4 operation invoc	eation from SP A to SP B.			
2. CHECK A: WAS THE INVOK	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
3. CHECK B: WAS THE REJECT	. CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?			
4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?				
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES				
Component portion in TSL messages				
Component portion tag: 01101100 Component portion length: correct number of octets				
INVOKE component in TSL message from SP A to SP B				
Component type tag: 10100001 (INVOKE) Component length: correct number of octets				
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)				

TEST NUMBER: 2.3.3.4 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN RESULT NOT-LAST component in TSL message from SP B to SP A

Component type tag: 10100111 (RETURN RESULT NOT-LAST)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Sequence tag: 00110000 (see Note)

Sequence length: correct number of octets (see Note)

Operation code tag: 00000010 (local) or 00000110 (global) (see Note)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)

Operation code: x (see Note)

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000010 (RETURN RESULT)

Problem code length: 00000001

Problem code: 00000001 (return result unexpected)

TEST NUMBER: 2.3.3.5			Sheet: 1 of 2
REFE	RENCE: 3.2.1/Q.774		
TITLI	E: Inopportune Behavior; Unexpec	cted Components	
SUBT	TTLE: Return Error for Class 3		
PURP	OSE: To verify that a rejection ca	n be sent if a Return Error component is received	d for a Class 3 operation
1) Ar		nat an appropriate TSL message generated at SP A	A contains an Invoke component
	CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP
EXPE	CTED MESSAGE AND COMPO	NENT FLOW:	
	SP A (CSL)		SP B (CSL)
	<i>TC-INVOKE req.</i> ====>		
	INVOKE (i)	>	
		<	RETURN ERROR (i)
	TC-L-REJECT ind. <=====		
	REJECT (i)	>	
TEST DESCRIPTION			
1.	Initiate a Class 3 operation invoca	ation from SP A to SP B.	
2.	CHECK A: WAS THE INVOK	E COMPONENT WITH CORRECT INFORMA	ATION SENT BY SP A?
3.	CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?		
4. CHECK C: WAS THE INVOCATION STATE MACHINE IDLE AT SP A?			
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages			
Component portion tag: 01101100 Component portion length: correct number of octets			
INVOKE component in TSL message from SP A to SP B			
Component type tag: 10100001 (INVOKE) Component length: correct number of octets			
Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.3.3.5 Sheet: 2 of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN ERROR) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000011 (RETURN ERROR)

Problem code length: 00000001 (unexpected return error)

TEST NUMBER: 2.3.3.6		Sheet: 1 of 2	
REFERENCE: 3.2.1/Q.774			
TITLE: Inopportune Behavior; Unexpected Co	omponents		
SUBTITLE: Return Error for Class 4			
PURPOSE: To verify that a rejection can be se	ent if a Return Error component is received	d for a Class 4 operation	
PRE-TEST CONDITIONS: 1) Arrange the TC-User stimulus such that an a 2) Arrange the data at SP B such that a Return		A contains an Invoke component	
CONFIGURATION: 1	TYPE OF TEST: VAT	TYPE OF SP: SP	
EXPECTED MESSAGE AND COMPONENT	FLOW:		
SP A (CSL)		SP B (CSL)	
TC-INVOKE req.			
INVOKE (i)	·····>		
TC-L-REJECT ind. <======	·····>	RETURN ERROR (i)	
TEST DESCRIPTION			
2. CHECK A: WAS THE INVOKE CO. 3. CHECK B: WAS THE REJECT CO.	CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A?		
CHECK TABLE FOR COMPONENTS WITHIN MESSAGES			
Component portion in TSL messages Component portion tag: 01101100 Component portion length: correct number of octets INVOKE component in TSL message from SP A to SP B Component type tag: 10100001 (INVOKE) Component length: correct number of octets Invoke ID tag: 00000010 Invoke ID length: 00000001 (one octet) Invoke ID: i (i represents an integer)			

TEST NUMBER: 2.3.3.6 Sheet: 2of 2

CHECK TABLE FOR COMPONENTS WITHIN MESSAGES

Operation code tag: 00000010 (local) or 00000110 (global)

Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)

Operation code: x (x represents a valid operation code)

parameters (provided by the TC-User)

RETURN ERROR component in TSL message from SP B to SP A

Component type tag: 10100011 (RETURN ERROR)

Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Error code tag: 00000010 (local) or 00000110 (global)

Error code length: correct number of octets (e.g. 00000001 if y is one octet long)

Error code: y

parameters (provided by the TC-User)

REJECT component in TSL message from SP A to SP B

Component type tag: 10100100 (REJECT) Component length: correct number of octets

Invoke ID tag: 00000010 Invoke ID length: 00000001

Invoke ID: i

Problem code tag: 10000011 (RETURN ERROR)

Problem code length: 00000001 (unexpected return error)