INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.787
(09/97)

SERIES Q: SWITCHING AND SIGNALLING

Specifications of Signalling System No. 7 – Test specification

# Transaction Capabilities (TC) test specification

ITU-T Recommendation Q.787

(Previously CCITT Recommendation)

ITU-T Q-SERIES RECOMMENDATIONS

**SWITCHING AND SIGNALLING**

| | |
|---|---|
| SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE | Q.1–Q.3 |
| INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING | Q.4–Q.59 |
| FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN | Q.60–Q.99 |
| CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS | Q.100–Q.119 |
| SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4 AND No. 5 | Q.120–Q.249 |
| SPECIFICATIONS OF SIGNALLING SYSTEM No. 6 | Q.250–Q.309 |
| SPECIFICATIONS OF SIGNALLING SYSTEM R1 | Q.310–Q.399 |
| SPECIFICATIONS OF SIGNALLING SYSTEM R2 | Q.400–Q.499 |
| DIGITAL EXCHANGES | Q.500–Q.599 |
| INTERWORKING OF SIGNALLING SYSTEMS | Q.600–Q.699 |
| SPECIFICATIONS OF SIGNALLING SYSTEM No. 7 | Q.700–Q.849 |
| General | Q.700 |
| Message transfer part (MTP) | Q.701–Q.709 |
| Signalling connection control part (SCCP) | Q.711–Q.719 |
| Telephone user part (TUP) | Q.720–Q.729 |
| ISDN supplementary services | Q.730–Q.739 |
| Data user part | Q.740–Q.749 |
| Signalling System No. 7 management | Q.750–Q.759 |
| ISDN user part | Q.760–Q.769 |
| Transaction capabilities application part | Q.770–Q.779 |
| **Test specification** | **Q.780–Q.799** |
| Q3 interface | Q.800–Q.849 |
| DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1 | Q.850–Q.999 |
| PUBLIC LAND MOBILE NETWORK | Q.1000–Q.1099 |
| INTERWORKING WITH SATELLITE MOBILE SYSTEMS | Q.1100–Q.1199 |
| INTELLIGENT NETWORK | Q.1200–Q.1999 |
| BROADBAND ISDN | Q.2000–Q.2999 |

*For further details, please refer to ITU-T List of Recommendations.*

**ITU-T RECOMMENDATION Q.787**

# TRANSACTION CAPABILITIES (TC) TEST SPECIFICATION

## Summary

This revised Recommendation Q.787 contains the test scripts for the SS No. 7 Transaction Capabilities. This revised Recommendation now covers the test descriptions for the Dialogue Portion of the *White Book* (1993) Recommendations Q.771 to Q.774 (Transaction capabilities application part).

## Source

ITU-T Recommendation Q.787 was revised by ITU-T Study Group 11 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 12th of September 1997.

NOTE

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

## CONTENTS

**Recommendation Q.787**

# TRANSACTION CAPABILITIES (TC) TEST SPECIFICATION

*(revised in 1997)*

## 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 *White Book* (1993) 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 *White Book* (1993) 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 subclauses: 7.1, TC Transaction Sublayer (TSL) test specification and 7.2, TC Component Sublayer (CSL) test specification. 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 IDs 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.

PRIMITIVE: ===========>

MESSAGE: $\longrightarrow$

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 7.1.1 and 7.2.1, Guidance on performing component sublayer tests.

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).

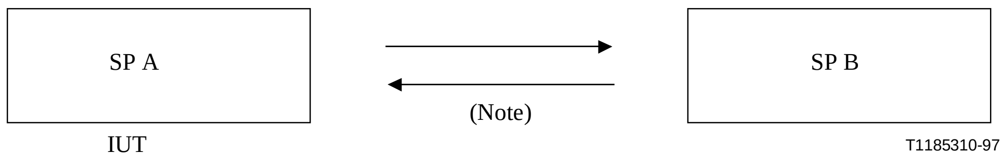
## 5 Test environment

### 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 signalling 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 (TSL) 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. If another message is received as a response, then this means that the transaction is not in the idle state.

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

### 7.1.2 Transaction sublayer tests

| | |
|---|---|
| NDA | No Details Available |
| FFS | For Further Study |
| * | Validation and Compatibility |

All other tests are Validation Only.

1 *Transaction sublayer*

  1.1 Valid function

    1.1.1 Unstructured dialogue

* 1.1.1.1 Tested side sending

* 1.1.1.2 Tested side receiving

    1.1.2 Structured dialogue

      1.1.2.1 Clearing before subsequent Message

        1.1.2.1.1 Valid clearing from initiating side

* 1) Prearranged ending

* 2) Abort by the TR-User

        1.1.2.1.2 Valid clearing 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

            2) Abort by transaction sublayer

* 3) Basic ending

      1.1.2.2 Clearing after Continue Message

        1.1.2.2.1 Valid clearing from initiating side

          1.1.2.2.1.1 IUT sending

* 1) Basic ending

* 2) Prearranged ending

* 3) Abort by the TR-User

          1.1.2.2.1.2 IUT receiving

* 1) Basic ending

            2) Abort by the transaction sublayer

* 3) Abort by the TR-User

1.1.2.2.2 Valid clearing from responding side

1.1.2.2.2.1 IUT sending

* 1) Basic ending

* 2) Prearranged ending

* 3) Abort by the TR-User

1.1.2.2.2.2 IUT receiving

* 1) Basic ending

* 2) Abort by the transaction sublayer

* 3) Abort by the TR-User

1.1.2.3 Clearing after Continue Message (component portion not present)

1.1.2.3.1 Basic ending IUT sending

1.1.2.3.2 Basic ending IUT receiving

1.1.2.4 Message exchange after transaction established

1.1.2.4.1 IUT initiating

1.1.2.4.2 IUT receiving

1.1.2.5 TC Addressing

1.1.2.5.1 Register address change

1.1.3 Encoding and value variations

1.1.3.1 Encoding variations

1.1.3.1.1 Length variations

1.1.3.1.1.1 Definite 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 Definite long

1) Component portion length in definite long form embedded in long form

1.1.3.1.1.3 Indefinite form

1) Component portion length in indefinite form embedded in indefinite form

1.1.3.2 Value variations

1.1.3.2.1 Transaction ID

* 1) Length is one octet

* 2) Length is four octets

1.2 Syntactically Invalid Behaviour

1.2.1 Invalid values for information elements

1.2.1.1 Begin Message type

1) OTID length = 0

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
3) OTID duplicated
4) 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 Inopportune 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 dialogue | | |
| SUBTITLE: Tested side sending | | |
| PURPOSE: To verify that signalling point A is able to correctly send a Unidirectional 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-UNI req.*
===========>

**UNIDIRECTIONAL** $\longrightarrow$

TEST DESCRIPTION

| | |
|---|---|
| 1. | Send a Unidirectional message from SP A to SP B. |
| 2. | CHECK A: WAS THE UNIDIRECTIONAL MESSAGE CORRECTLY SENT FROM 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: 1.1.1.2 | | 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 | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT and CPT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| | ←―――――――――――――――― | **UNIDIRECTIONAL** |
| *TR-UNI ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Send a Unidirectional message from SP B to SP A. |
| 2. | 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 SP A. |
| 3. | CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM 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
Component portion length: correct number of octets

| | | |
|---|---|---|
| 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.*<br>============> | | |
| **BEGIN** | $\longrightarrow$ | |
| *TR-U-ABORT req.*<br>============> | | |

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-U-ABORT request primitive 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 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

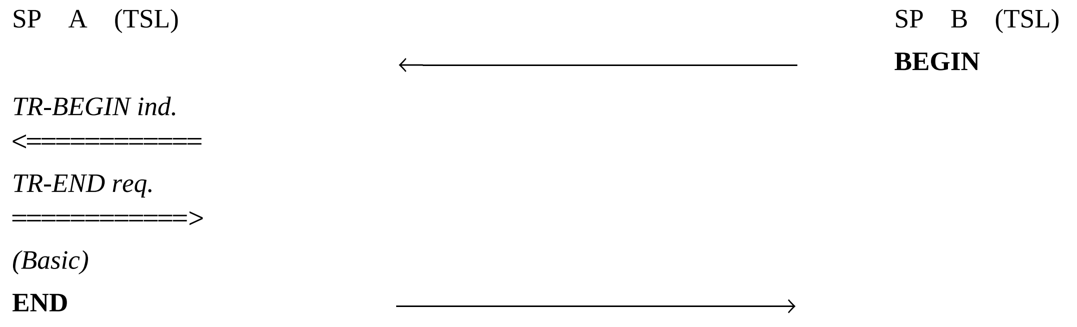
Message type tag: 01100010
Message type length: correct number of octets

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
Component portion length: correct number of octets

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.1.2.1 1) | | Sheet: 1 of 2 |
| 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; Basic ending | | |
| PURPOSE: To verify that signalling point A is able to receive a Begin 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 |

EXPECTED MESSAGE SEQUENCE:



TEST DESCRIPTION

| | |
|---|---|
| 1. | Send a Begin message from SP B to SP A. |
| 2. | On receipt of BEGIN indication arrange for a TR-END request primitive (basic) to be passed to the TSL at SP A. |
| 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 C: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.1.2.1.2.1 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| 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.*<br><============ | | |
| *TR-END req.*<br>============><br>*(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
Component portion length: correct number of octets

| | |
|---|---|
| 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.*<br><============ | | |
| *TR-U-ABORT req.*<br>============> | | |
| **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. |
| 3. | CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A? |
| 4. | CHECK B: WAS AN ABORT MESSAGE CORRECTLY SENT BY SP A? |
| 5. | CHECK C: WAS THE DTID IN THE ABORT MESSAGE THE SAME AS THE OTID IN THE BE GIN MESSAGE? |
| 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
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.* <br> ===========> | | |
| **BEGIN** | $\longrightarrow$ | |
| | $\longleftarrow$ | **ABORT (U)** |
| *TR-U-ABORT ind.* <br> <=========== | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Send a Begin message from SP A to SP B. |
| 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? |
| 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 (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 sublayer | | |
| 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 A (TSL) SP B (TSL)

*TR-BEGIN req.*
============>

**BEGIN** $\longrightarrow$

$\longleftarrow$ **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. |
| 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 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 NUMBER: 1.1.2.1.2.2 3) | | 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 receiving; Basic ending | | |
| PURPOSE: To verify that the signalling point A is able to terminate a transaction on reception of an END 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** | ⟶ | |
| | ⟵ | **END** |
| *TR-END ind.* <============ | | |
| TEST 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 STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? | |
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.2.1.1 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 initiating side; IUT sending; Basic ending | | |
| PURPOSE: To verify that the signalling point A is able to 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. 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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.*<br>============> | | |
| **BEGIN** | ——————————→ | |
| | ←—————————— | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |
| *TR-END req.*<br>============> | | |
| *(Basic)* | | |
| **END** | ——————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 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-END request primitive (basic) 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 CONTINUE MESSAGE CORRECTLY RECEIVED BY THE TSL AT SP A? |
| 5. | CHECK C: WAS THE END MESSAGE CORRECTLY SENT BY SP A? |
| 6. | CHECK D: WAS THE DTID IN THE END MESSAGE THE SAME AS THE OTID IN THE CONTINUE MESSAGE? |
| 7. | CHECK E: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.1.2.2.1.1 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in CONTINUE message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.2.1.1 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 | | |
| SUBTITLE: Clearing after continue Message; Valid clearing from initiating side; IUT sending; Prearranged ending | | |
| PURPOSE: To verify that signalling point A is able to 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. 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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.* | | |
| ============> | | |
| **BEGIN** | $\longrightarrow$ | |
| | $\longleftarrow$ | **CONTINUE** |
| *TR-CONTINUE ind.* | | |
| <============ | | |
| *TR-END req.* | | |
| ============> | | |
| *(Prearranged)* | | |

TEST DESCRIPTION

| | |
|---|---|
| 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-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 CONTINUE MESSAGE CORRECTLY RECEIVED AT SP A? |
| 5. | 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. |
| 6. | CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.1.2.2.1.1 2) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.2.1.1 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 the signalling point A is able to terminate the transaction by the "abort" method | | |
| 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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.* <br> ===========> | | |
| **BEGIN** | $\longrightarrow$ | |
| | $\longleftarrow$ | **CONTINUE** |
| *TR-CONTINUE ind.* <br> <=========== | | |
| *TR-U-ABORT req.* <br> ===========> | | |
| **ABORT (U)** | $\longrightarrow$ | |

TEST DESCRIPTION

| | |
|---|---|
| 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 CONTINUE MESSAGE CORRECTLY RECEIVED AT SP A? |
| 5. | CHECK C: WAS THE ABORT MESSAGE CORRECTLY SENT BY SP A? |
| 6. | CHECK D: WAS THE DTID IN THE ABORT MESSAGE THE SAME AS THE OTID IN THE CONTINUE MESSAGE? |
| 7. | CHECK E: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| 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

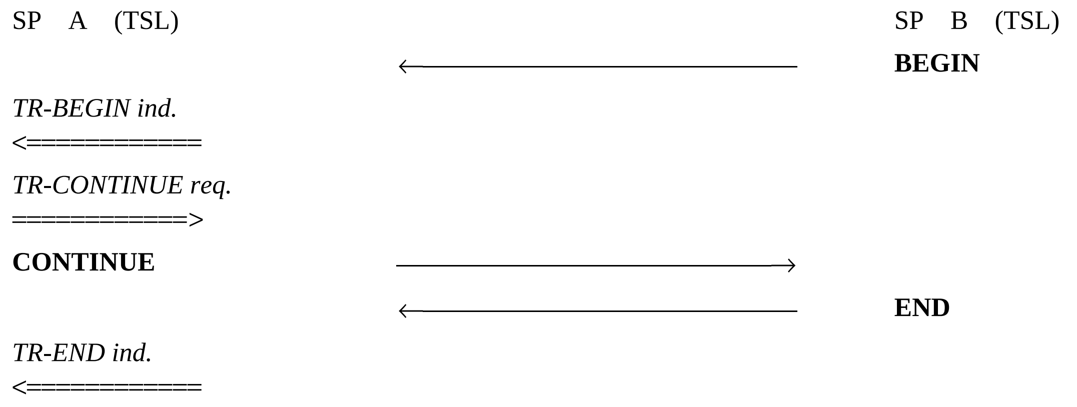
Message type tag: 01100111
Message type length: correct number of octets

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 |
| REFERENCE: 3.3.3.2.1.2/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 initiating side; IUT receiving; Basic ending | | |
| PURPOSE: To verify that signalling point A is able to generate a Continue message and then terminate the transaction on reception of an End 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:



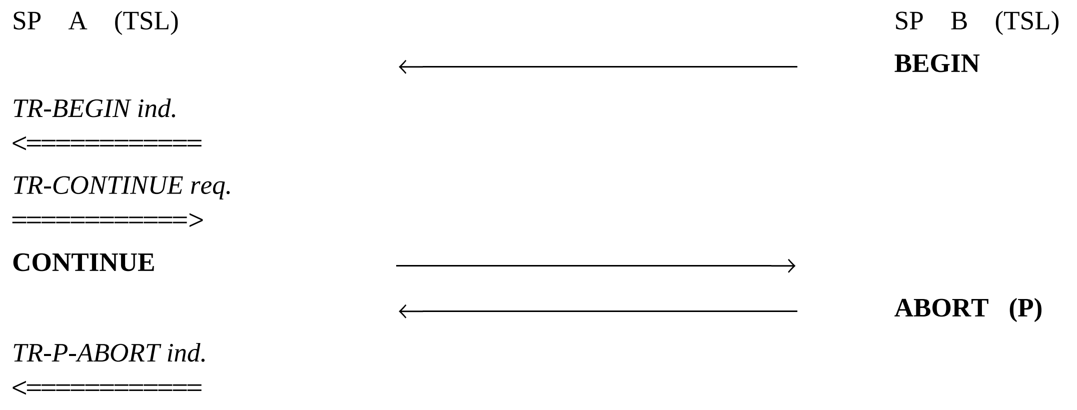
TEST DESCRIPTION

| | |
|---|---|
| 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 MESSAGE CORRECTLY RECEIVED AT SP A? |
| 5. | CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY SENT FROM SP A? |
| 6. | CHECK C: WAS THE END 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 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.2.1.2 2) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.3.2.1.2/Q.774 and 3.3.4/Q.774 | | |
| TITLE: Valid function; Structured dialogue | | |
| SUBTITLE: Clearing after continue Message; Valid clearing from initiating side; IUT receiving; Abort by the transaction sublayer | | |
| 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 TSL | | |
| 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:



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 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 2) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: one octet<br>P-Abort cause value: INTEGER (0 .. 4) | |

| | | |
|---|---|---|
| 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 | | |
| TITLE: Valid function; Structured dialogue | | |
| SUBTITLE: Clearing after continue Message; Valid clearing from initiating side; IUT receiving; 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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| | ⟵ | **BEGIN** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE** | ⟶ | |
| | ⟵ | **ABORT (U)** |
| *TR-U-ABORT ind.*<br><============ | | |

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<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (U)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br>User abort information tag: 01101011<br>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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| | ←―――――――――――――― | **BEGIN** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE** | ――――――――――――――→ | |
| *TR-END req.*<br>============> | | |
| *(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

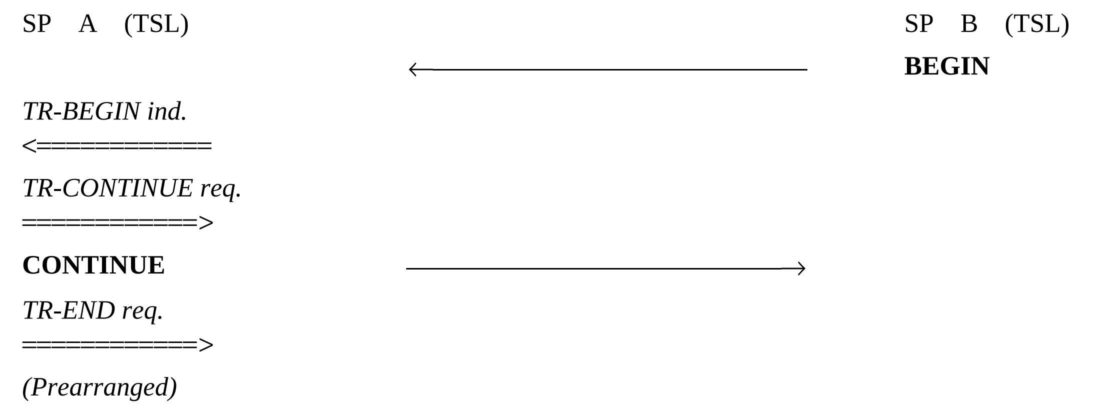
Destination transaction ID tag: 01001001
Destination transaction ID length: correct number of octets
Destination transaction ID value: OCTET STRING (1-4 octets long)
(OTID value received in CONTINUE message)

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

| | |
|---|---|
| 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 | |
| SUBTITLE: 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 |

EXPECTED MESSAGE SEQUENCE:



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<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| 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 CPT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| | ←―――――――――――――― | **BEGIN** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE** | ――――――――――――――→ | |
| *TR-U-ABORT req.*<br>============> | | |
| **ABORT (U)** | ――――――――――――――→ | |

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 STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.1.2.2.2.1 3) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in CONTINUE message)<br>User abort information tag: 01101011<br>User abort information length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.2.2.2 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.3.2.1/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 receiving; Basic ending | | |
| PURPOSE: To verify that the signalling point A is able to terminate the transaction on reception of an End message following a Continue 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.*<br>============> | | |
| **BEGIN** | ——————————————→ | |
| | ←—————————————— | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |
| | ←—————————————— | **END** |
| *TR-END ind.*<br><============ | | |

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 CONTINUE MESSAGE CORRECTLY RECEIVED AT SP A? |
| 6. | CHECK C: WAS THE END 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: 1.1.2.2.2.2 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.2.2.2 2) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.3.2.1/Q.774 and 3.3.4/Q.774 | | |
| TITLE: Valid function; Structured dialogue | | |
| SUBTITLE: Clearing after continue Message; Valid clearing from responding side; IUT receiving; Abort by the transaction sublayer | | |
| PURPOSE: To verify that the signalling point A is able to terminate the transaction on reception of an Abort (P) message following a Continue 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 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 CONTINUE MESSAGE CORRECTLY RECEIVED AT 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: 1.1.2.2.2.2 2) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: one octet<br>P-Abort cause value: INTEGER (0 .. 4) | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.2.2.2 3) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.3.2.1/Q.774 and 3.3.3.2.4/Q.774 | | |
| TITLE: Valid function; Structured dialogue | | |
| SUBTITLE: Clearing after continue 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 the transaction on reception of an Abort (U) message following a Continue 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.*<br>============> | | |
| **BEGIN** | ⟶ | |
| | ⟵ | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |
| | ⟵ | **ABORT (U)** |
| *TR-U-ABORT ind.*<br><============ | | |

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 (U) message from SP B. |
| 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: 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: 1.1.2.2.2.2 3) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (U)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>User abort information tag: 01101011<br>User abort information length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.3.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.3/Q.774 | | |
| TITLE: Valid function; Structured dialogue | | |
| SUBTITLE: Clearing after continue Message (component portion not present); Basic ending IUT sending | | |
| PURPOSE: To verify that SP A is able to accept a Continue message without CP | | |
| 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.* <============ | | |
| *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 Continue 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 CONTINUE MESSAGE CORRECTLY RECEIVED AT SP A? |
| 6. | CHECK C: WAS THE END MESSAGE CORRECTLY SENT FROM SP A? |
| 7. | CHECK D: WAS THE TSL STATE 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<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets (range 1-4)<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets (range 1-4)<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets (range 1-4)<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets (range 1-4)<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.3.2 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.3/Q.774 | | |
| TITLE: Valid function; Structured dialogue | | |
| SUBTITLE: Clearing after continue Message (component portion not present); Basic ending IUT receiving | | |
| PURPOSE: To verify that SP A is able 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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| | ←―――――――――――――― | **BEGIN** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE** | ――――――――――――――→ | |
| | ←―――――――――――――― | **END** |
| *TR-END ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a BEGIN message to SP A without CP. |
| 2. | Arrange for SP A to send a CONTINUE 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 CONTINUE MESSAGE CORRECTLY SENT FROM SP A? |
| 6. | CHECK C: WAS THE END MESSAGE CORRECTLY RECEIVED AT SP A? |
| 7. | CHECK D: 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<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets (range 1-4)<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets (range 1-4)<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets (range 1-4)<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets (range 1-4)<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in CONTINUE message) | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.4.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.3/Q.774 | | |
| TITLE: Valid function; Structured dialogue | | |
| SUBTITLE: Message exchange after transaction established; IUT initiating | | |
| PURPOSE: To verify the correct message flow between SP A and SP B, after transaction 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.*<br>============> | | |
| **BEGIN** | ——————————→ | |
| | ←—————————— | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE** | ——————————→ | |
| | ←—————————— | **END** |
| *TR-END ind.*<br><============ | | |

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 to SP A. |
| 3. | Arrange for SP A to send a Continue message to SP B. |
| 4. | Arrange for SP B to send an END message to SP A. |
| 5. | CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? |
| 6. | CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY RECEIVED AT SP A? |
| 7. | CHECK C: WAS THE CONTINUE MESSAGE CORRECTLY SENT FROM SP A? |
| 8. | CHECK D: WAS THE END MESSAGE CORRECTLY RECEIVED AT SP A? |
| 9. | 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<br>Message type tag:   01100010<br>Message type length:   correct number of octets<br>Originating transaction ID tag:   01001000<br>Originating transaction ID length:   correct number of octets (range 1-4)<br>Originating transaction ID value:   OCTET STRING (1-4 octets long)<br>Component portion tag:   01101100<br>Component portion length:   correct number of octets<br>CONTINUE<br>Message type tag:   01100101<br>Message type length:   correct number of octets<br>Originating transaction ID tag:   01001000<br>Originating transaction ID length:   correct number of octets (range 1-4)<br>Originating transaction ID value:   OCTET STRING (1-4 octets long)<br>Destination transaction ID tag:   01001001<br>Destination transaction ID length:   0 (invalid length)<br>Destination transaction ID value:   OCTET STRING (1-4 octets long) (OTID received in BEGIN message)<br>Component portion tag:   01101100<br>Component portion length:   correct number of octets<br>CONTINUE<br>Message type tag:   01100101<br>Message type length:   correct number of octets<br>Originating transaction ID tag:   01001000<br>Originating transaction ID length:   correct number of octets (range 1-4)<br>Originating transaction ID value:   OCTET STRING (1-4 octets long)<br>Destination transaction ID tag:   01001001<br>Destination transaction ID length:   0 (invalid length)<br>Destination transaction ID value:   OCTET STRING (1-4 octets long) (OTID value received in CONTINUE message)<br>Component portion tag:   01101100<br>Component portion length:   correct number of octets<br>END<br>Message type tag:   01100101<br>Message type length:   correct number of octets<br>Destination transaction ID tag:   01001001<br>Destination transaction ID length:   0 (invalid length)<br>Destination transaction ID value:   OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>Component portion tag:   01101100<br>Component portion length:   correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.4.2 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.3/Q.774 | | |
| TITLE: Valid function; Structured dialogue | | |
| SUBTITLE: Message exchange after transaction established; IUT receiving | | |
| PURPOSE: To verify the correct message flow between SP A and SP B, after transaction 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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| | ←―――――――――――― | **BEGIN** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE** | ――――――――――――→ | |
| | ←―――――――――――― | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |
| *TR-END req.*<br>============> | | |
| **END** | ――――――――――――→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message to SP A. |
| 2. | Arrange for SP A to send a Continue message to SP B. |
| 3. | Arrange for SP B to send a Continue message to SP A. |
| 4. | Arrange for SP A to send an END message to SP B. |
| 5. | CHECK A: WAS THE BEGIN MESSAGE CORRECTLY RECEIVED AT SP A? |
| 6. | CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY SENT FROM SP A? |
| 7. | CHECK C: WAS THE CONTINUE MESSAGE CORRECTLY RECEIVED AT 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
Component portion length: correct number of octets

| | | |
|---|---|---|
| TEST NUMBER: 1.1.2.5.1 | | Sheet: 1 of 1 |
| REFERENCE: 3.1.2.2.2.2/Q.771 | | |
| TITLE: Valid function, Structured dialogue | | |
| SUBTITLE: TC addressing; Register address change | | |
| PURPOSE: To verify that a correctly reported address change of the peer implementation is registered and used in subsequent messages. | | |
| 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:

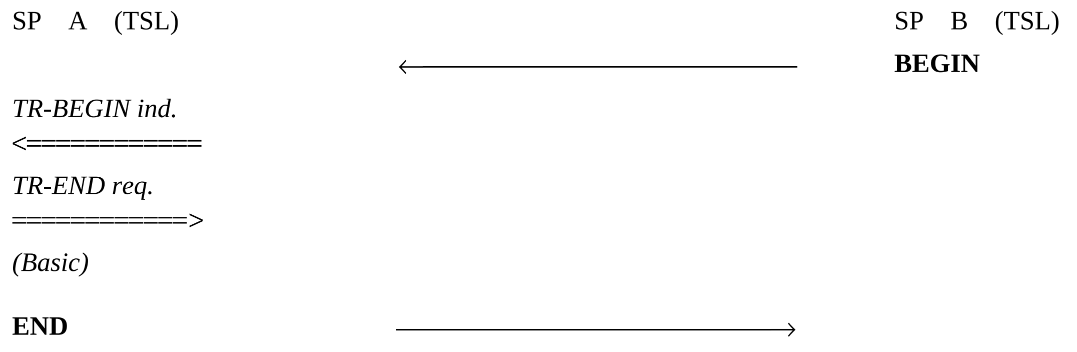
| | | |
|---|---|---|
| SP A (TSL) | | SP B (TSL) |
| | UDT ( CLG A, CLD B) | |
| **BEGIN** | ⟶ | |
| | | *TR-BEGIN ind.*<br>============> |
| | | *TR-CONTINUE req.*<br><============ |
| | UDT ( CLG B*, CLD A) | |
| | ⟵ | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |
| *TR-END req*<br>============> | | |
| | UDT ( CLG A, CLD B*) | |
| **END** | ⟶ | |

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 in which the calling address has been optimized. |
| 3. | Arrange for SP A to respond with an End message. |
| 4. | CHECK A: WAS THE CALLED ADDRESS IN THE SCCP MESSAGE HEADER FOR THE END THE SAME AS THE CALLING ADDRESS IN THE SCCP MESSAGE HEADER FOR THE CONTINUE MESSAGE? |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.3.1.1.1 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3/Q.774 | | |
| TITLE: Valid function; Encoding and value variations | | |
| SUBTITLE: Encoding variations; Length variations; Definite short; Component portion length in definite short form embedded in short form | | |
| 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 (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:



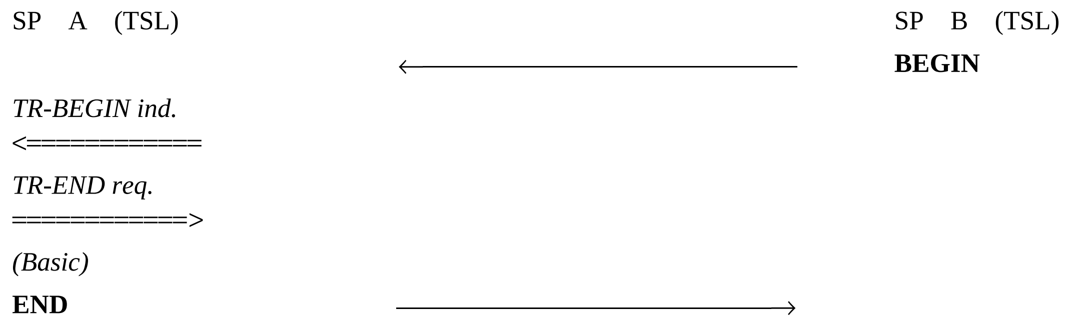
TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message to SP A with lengths encoded as described 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 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: 1.1.3.1.1.1 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets coded in definite short form<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: one octet<br>Originating transaction ID value: OCTET STRING (1 octet)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets coded in definite short form<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: one octet<br>Destination transaction ID value: OCTET STRING (1 octet)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.3.1.1.1 2) | | Sheet: 1 of 2 |
| REFERENCE: 3.3/Q.774 | | |
| TITLE: Valid function; Encoding and value variations | | |
| SUBTITLE: Encoding variations; Length variations; Definite short; Component portion length in definite short form embedded in long form | | |
| PURPOSE: To verify that signalling point A is able to accept a Begin message whose length is encoded using the definite long form and with a component portion whose length is encoded using the definite short form | | |
| 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:



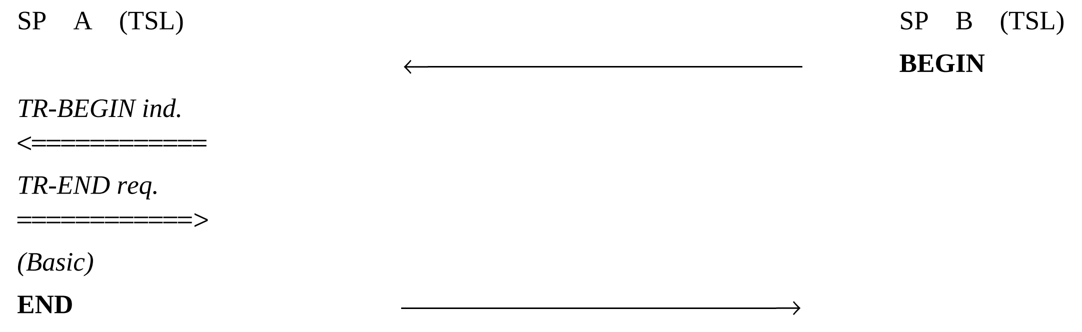
TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message to SP A with lengths encoded as described 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 MESSAGE CORRECTLY SENT BY SP A? |
| 5. | CHECK C: WAS THE DTID IN THE END MESSAGE THE SAME AS THE OTID IN THE BE GIN MESSAGE? |
| 6. | CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.1.3.1.1.1 2) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets coded in definite long form<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: one octet<br>Originating transaction ID value: OCTET STRING (1 octet)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets coded in definite short form<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: one octet<br>Destination transaction ID value: OCTET STRING (1 octet)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.3.1.1.2 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3/Q.774 | | |
| TITLE: Valid function; Encoding and value variations | | |
| SUBTITLE: Encoding variations; Length variations; Definite long; Component portion length in definite long form embedded in long form | | |
| PURPOSE: To verify that signalling point A is able to accept a Begin message whose length is encoded using the definite long form and with a component portion whose length is encoded using the definite long form | | |
| 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:



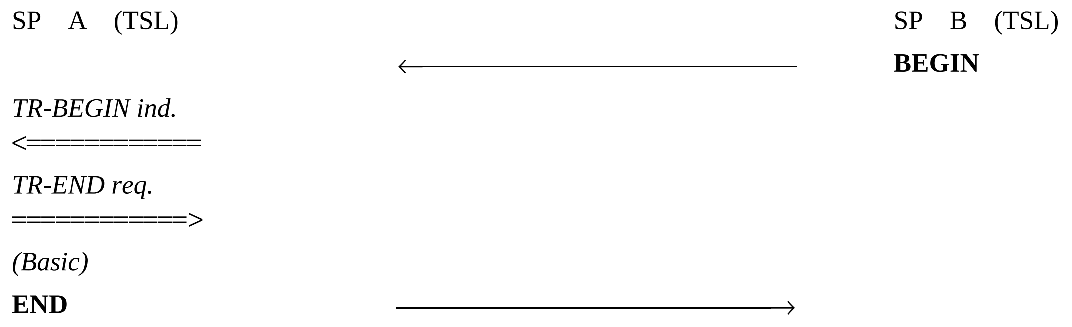
TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message to SP A with lengths encoded as described 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 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: 1.1.3.1.1.2 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets coded in definite long form<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: one octet<br>Originating transaction ID value: OCTET STRING (1 octet)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets coded in definite long form<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: one octet<br>Destination transaction ID value: OCTET STRING (1 octet) (OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.3.1.1.3 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3/Q.774 | | |
| TITLE: Valid function; Encoding and value variations | | |
| SUBTITLE: Encoding variations; Length variations; Indefinite form; Component portion length in indefinite form embedded in indefinite form | | |
| PURPOSE: To verify that signalling point A is able to accept a Begin message whose length is encoded using the indefinite form and with a component portion whose length is encoded using the indefinite form | | |
| 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:



TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message to SP A with lengths encoded as described 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 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: 1.1.3.1.1.3 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br><br>Message type tag: 01100010<br>Message type length: correct number of octets coded in indefinite form<br><br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: one octet<br>Originating transaction ID value: OCTET STRING (1 octet)<br><br>Component portion tag: 01101100<br>Component portion length: correct number of octets coded in indefinite form<br>Component contents provided by TC user<br><br>EOC Tag: 00000000, Length: 00000000<br><br>END<br><br>Message type tag: 01100100<br>Message type length: correct number of octets<br><br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: one octet<br>Destination transaction ID value: OCTET STRING (1 octet)<br>(OTID value received in BEGIN message)<br><br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.3.2.1 1) | | Sheet: 1 of 2 |
| REFERENCE: 5.3/Q.774 | | |
| TITLE: Valid function; Encoding and value variations | | |
| SUBTITLE: Value variations; Transaction ID; Length is one octet | | |
| PURPOSE: To verify that signalling point A is able to deal with correct encoding of OTID information element (1 octet long) | | |
| 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 |

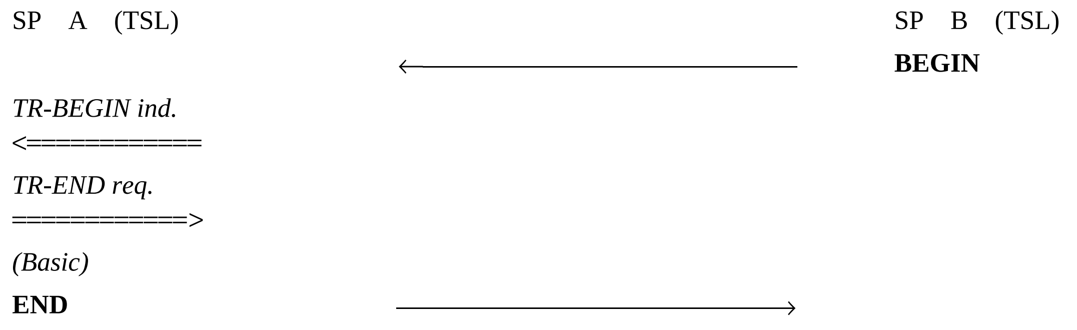
EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| | ←―――――――――――――― | **BEGIN** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-END req.*<br>============> | | |
| *(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<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: one octet<br>Originating transaction ID value: OCTET STRING (1 octet)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: one octet<br>Destination transaction ID value: OCTET STRING (1 octet) (OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.1.3.2.1 2) | | Sheet: 1 of 2 |
| REFERENCE: 5.3/Q.774 | | |
| TITLE: Valid function; Encoding and value variations | | |
| SUBTITLE: Value variations; Transaction ID; Length is four octets | | |
| PURPOSE: To verify that signalling point A is able to deal with correct encoding of OTID information element (4 octets long) | | |
| 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)<br>SP B (TSL)<br>**BEGIN**<br>*TR-BEGIN ind.*<br><============<br>*TR-END req.*<br>============><br>*(Basic)*<br>**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 STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.1.3.2.1 2) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: four octets<br>Originating transaction ID value: OCTET STRING (4 octets)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: four octets<br>Destination transaction ID value: OCTET STRING (4 octets)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | |
|---|---|
| TEST NUMBER: 1.2.1.1 1) | Sheet: 1 of 1 |
| REFERENCE: 3.3.4/Q.774 | |
| TITLE: Syntactically invalid behaviour; 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* | $\longleftarrow$ | **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
Component portion length: correct number of octets

| | |
|---|---|
| TEST NUMBER: 1.2.1.1 2) | Sheet: 1 of 1 |
| REFERENCE: 3.3.4/Q.774 | |
| TITLE: Syntactically invalid behaviour; 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* | $\longleftarrow$ | **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. |
| 5. | CHECK D: 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 octets
Originating transaction ID value: OCTET STRING (5 octets long)

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

| | | |
|---|---|---|
| TEST NUMBER: 1.2.1.2 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid values for information elements | | |
| SUBTITLE: First continue Message; DTID length = 0 | | |
| PURPOSE: To verify that on receipt of a corrupted Continue message, with DTID length = 0, SP A is able to discard the message or abort the transaction 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 that the first Continue message contains a DTID of length = 0 | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN** | $\longrightarrow$ | |
| *Detect syntax error* | $\longleftarrow$ | **CONTINUE** |
| **ABORT (P)** (see Note) | $\longrightarrow$ | |

NOTE – If the Abort is not sent, this may be valid behaviour 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. |
| 3. | CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? |
| 4. | CHECK B: VERIFY THAT THE TR-USER AT SP A WAS NOT INFORMED OF THE CONTINUE MESSAGE AT SP A. |
| 5. | CHECK C: WERE THE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION PRIOR TO THE CONTINUE MESSAGE LEFT IN INITIATION SENT STATE ? |
| 6. | CHECK D: IF AN ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH CORRECT DTID AND P-ABORT CAUSE VALUE? |

| TEST NUMBER: 1.2.1.2 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: correct number of octets<br>P-Abort cause value: incorrect transaction portion | |

| | | |
|---|---|---|
| TEST NUMBER: 1.2.1.3 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid values for information elements | | |
| SUBTITLE: Subsequent continue Message; Component portion length incorrect | | |
| PURPOSE: To verify that on receipt of a corrupted Continue message with OTID derivable and DTID derivable and assigned, after transaction establishment, SP A is able to abort the transaction | | |
| 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.*<br>===========> | | |
| **BEGIN** | ——————————→ | |
| | ←—————————— | **CONTINUE** |
| *TR-CONTINUE ind.*<br><=========== | | |
| *Detect error* | ←—————————— | **CONTINUE** |
| **ABORT (P)** (see Note) | ——————————→ | |
| *TR-P-ABORT ind.*<br><=========== | | |

NOTE – If the Abort is not sent, this may be valid behaviour depending on the implementation.

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 WAS SENT, WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.2.1.3 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE (1st)<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE (2nd)<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: correct number of octets<br>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 behaviour; 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.*<br>===========> | | |
| **BEGIN** | ⟶ | |
| *Detect 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 (invalid DTID length). |
| 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 END MESSAGE. |
| 5. | 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
Component portion length: correct number of octets

| | | |
|---|---|---|
| TEST NUMBER: 1.2.1.5 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid values for information elements | | |
| SUBTITLE: Abort message; Invalid P-Abort cause value | | |
| PURPOSE: To verify that signalling point A is able to deal with incorrect encoding of P-Abort cause information element (illegal value) | | |
| 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 Abort message with a DTID that is derivable and assigned, contains a syntax error and is sent to SP A in response to the Begin message | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.*<br>============> | | |
| **BEGIN** | ——————————→ | |
| *Detect syntax error* | ←—————————— | **ABORT (P)** |
| *TR-P-ABORT ind.*<br><============ | | |

NOTE – The sending of the TR-Abort ind. is implementation dependent.

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 NO MESSAGES ARE GENERATED BY SP A IN RESPONSE TO THE CORRUPTED ABORT MESSAGE. |
| 4. | CHECK C: IF THE TR-ABORT IND. WAS SENT, WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.2.1.5 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: correct number of octets<br>P-Abort cause value: INTEGER (5 – Illegal value for this field) | |

| | |
|---|---|
| TEST NUMBER: 1.2.1.5 2) | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | |
| TITLE: Syntactically invalid behaviour; Invalid values for information elements | |
| SUBTITLE: Abort message; P-Abort cause length incorrect | |
| PURPOSE: To verify that on receipt of a corrupted Abort message with incorrect cause length, signalling point A is able to discard the message and advise the local user | |
| 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 Abort message with a DTID that is derivable and assigned, contains a syntax error and is sent to SP A in response to the Begin message | |

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

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN** | ⟶ | |
| *Detect syntax error* | ⟵ | **ABORT (P)** |
| *TR-P-ABORT ind.*<br><=========== | | |

NOTE – The sending of the TR-Abort ind. is implementation dependent.

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. (Corrupted P-Abort cause length). |
| 2. | CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT FROM SP A? |
| 3. | CHECK B: VERIFY THAT NO MESSAGES ARE GENERATED BY SP A IN RESPONSE TO THE CORRUPTED ABORT MESSAGE. |
| 4. | CHECK C: IF THE TR-ABORT IND. WAS SENT, WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.2.1.5 2) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: correct number of octets (i.e. not one)<br>P-Abort cause value: INTEGER (0 .. 4) | |

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.1 1) | | Sheet: 1 of 1 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Unidirectional Message type; Unknown information element present | | |
| PURPOSE: To verify that on receipt of a corrupted Unidirectional 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 Unidirectional message contains a syntax error and is sent to SP A | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *Detect syntax error* | ⟵ | **UNIDIRECTIONAL** |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send the corrupted Unidirectional message to SP A. |
| 2. | CHECK A: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE UNIDIRECTIONAL MESSAGE AT SP A. |
| 3. | CHECK B: VERIFY THAT NO MESSAGES WERE GENERATED IN RESPONSE TO THE UNIDIRECTIONAL 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

UNIDIRECTIONAL

Message type tag: 01100001
Message type length: correct number of octets

Component portion missing

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.2 1) | | Sheet: 1 of 1 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Begin Message type; OTID absent | | |
| 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 a syntax error and the OTID is not derivable | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *Detect syntax error* | $\longleftarrow$ | **BEGIN** |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send the corrupted Begin message to SP A, with OTID not present. |
| 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 IN RESPONSE TO THE THE CORRUPTED BEGIN 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

BEGIN

Message type tag: 01100010
Message type length: correct number of octets

OTID absent

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.2 2) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Begin Message type; Unknown information element present | | |
| PURPOSE: To verify that on receipt of a corrupted Begin message, with an invalid information element, signalling point A is able to discard the message and generate an Abort 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 a syntax error | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *Detect syntax error* | ← | **BEGIN** |
| **ABORT (P)** | → | |

NOTE – If the Abort is not sent, this may be valid behavior depending on the implementation.

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send the corrupted Begin message to SP A, with an invalid information 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. | CHECK B: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.2.2.2 2) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br><br>Message type tag: 01100010<br>Message type length: correct number of octets<br><br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br><br>Information element tag: unknown (eg. 01101101)<br>Information element length: correct number of octets<br>Information element value: OCTET STRING<br><br>ABORT (P)<br><br>Message type tag: 01100111<br>Message type length: correct number of octets<br><br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br><br>P-Abort cause tag: 01001010<br>P-Abort cause length: correct number of octets<br>P-Abort cause value: incorrect transaction portion 00000011 | |

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.3 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: First Continue Message; OTID absent | | |
| PURPOSE: To verify that on receipt of a corrupted Continue 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. SP B to respond with a Continue message on receipt of the Begin message. Arrange the data at SP B such that the Continue message contains a syntax error and the OTID is not derivable | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.* ===========> | | |
| **BEGIN** | $\longrightarrow$ | |
| *Detect syntax error* | $\longleftarrow$ | **CONTINUE** |

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 (OTID not derivable) 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. | CHECK C: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE CORRUPTED CONTINUE MESSAGE? |
| 6. | CHECK D: WERE TSL STATE MACHINES ASSOCIATED WITH THE TRANSACTION, PRIOR TO THE CONTINUE MESSAGE, LEFT IN THE INITIATION SENT STATE AT SP A? |

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

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.3 2) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: First Continue Message; DTID absent | | |
| PURPOSE: To verify that on receipt of a corrupted Continue message containing no DTID, signalling point A is able to discard the message or abort the transaction | | |
| 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 the Continue message contains no DTID | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.* ===========> | | |
| **BEGIN** | $\longrightarrow$ | |
| *Detect syntax error* | $\longleftarrow$ (DTID absent) | **CONTINUE** |
| **ABORT (P)** | $\longrightarrow$ | |

NOTE – If the Abort is not sent, this may be valid behaviour depending on the implementation.

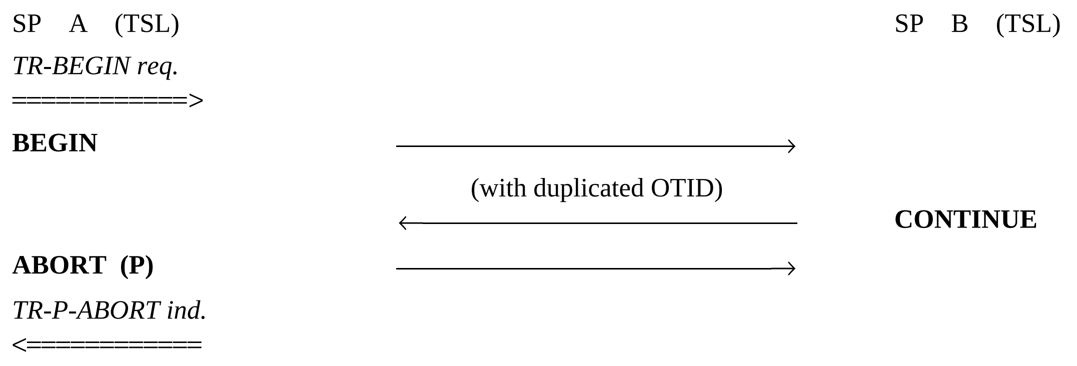
TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message to SP B. |
| 2. | Arrange for SP B to send the corrupted Continue message (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 CONTINUE MESSAGE AT SP A. |
| 5. | CHECK C: WERE THE TSL STATE MACHINES ASSOCIATED WITH THE TRANSACTION, PRIOR TO THE CONTINUE MESSAGE, LEFT IN THE INITIATION SENT STATE? |
| 6. | CHECK D: IF AN ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH CORRECT DTID AND CORRECT P-ABORT CAUSE VALUE? |

| TEST NUMBER: 1.2.2.3 2) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>DTID absent<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: correct number of octets<br>P-Abort cause value: incorrect transaction portion 00000011 | |

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.3 3) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: First Continue Message; OTID duplicated | | |
| PURPOSE: To check the correct behaviour of the implementation under test on receipt of a first Continue message with a duplicated OTID | | |
| 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:



NOTE – If the ABORT message and primitive are not sent, this may be valid behaviour depending on the implementation.

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange SP A to send a Begin message. |
| 2. | Arrange for SP B to send a Continue 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 MESSAGE AND PRIMITIVE WERE SENT, WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE 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 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: First Continue Message; DTID duplicated | | |
| PURPOSE: To check the correct behaviour of the implementation under test on receipt of a first Continue message with a duplicated DTID | | |
| 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** | ⟶ | |
| | (with duplicated DTID) ⟵ | **CONTINUE** |
| **ABORT (P)** | ⟶ | |
| *TR-P-ABORT ind.* <=========== | | |

NOTE – If the ABORT message and primitive are not sent, this may be valid behaviour depending on the implementation.

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange SP A to send a Begin message. |
| 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 MESSAGE AND PRIMITIVE WERE SENT, WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE 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 |
| REFERENCE: 3.3.4/Q.774 | |
| TITLE: Syntactically invalid behaviour; Invalid structure | |
| SUBTITLE: First Continue Message; Unknown information element present | |
| PURPOSE: To verify that on receipt of a corrupted Continue message, 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 such that a Continue message with an OTID that is derivable and a DTID that is derivable and assigned, contains a syntax error and is sent to SP A in response to the Begin message | |

| | | |
|---|---|---|
| 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* | ←—————————— | **CONTINUE** |
| **ABORT (P)** | ——————————→ | |
| *TR-P-ABORT ind.* <=========== | | |

NOTE – If the ABORT message and primitive are not sent, this may be valid behaviour 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 with an extra information element after the DTID information element (eg P-Abort Cause). |
| 3. | CHECK A: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE CONTINUE MESSAGE AT SP A. |
| 4. | CHECK B: IF THE ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A, WITH CORRECT DTID AND THE CORRECT P-ABORT CAUSE VALUE? (INCORRECT TRANSACTION PORTION) |
| 5. | CHECK C: IF THE MESSAGE AND PRIMITIVE ABORT WERE SENT, WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.2.2.3 5) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>Information element tag: unknown (eg. 01101101)<br>Information element length: correct number of octets<br>Information element value: OCTET STRING<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in CONTINUE message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: correct number of octets<br>P-Abort cause value: incorrect transaction portion 00000011 | |

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.4 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Subsequent Continue Message; OTID absent | | |
| PURPOSE: To verify that on receipt of a corrupted Continue message after transaction establishment, 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 | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN** | ⟶ | |
| | ⟵ | **CONTINUE** |
| *TR-CONTINUE ind.*<br><=========== | | |
| *Detect error* | ⟵ | **CONTINUE** |

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 (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. |
| 7. | CHECK D: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE CORRUPTED CONTINUE MESSAGE. |
| 8. | CHECK E: WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION, PRIOR TO THE CORRUPTED CONTINUE MESSAGE, LEFT IN THE ACTIVE STATE AT SP A? |

| TEST NUMBER: 1.2.2.4 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE (1st)<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) (OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE (2nd)<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>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 |
|---|---|

REFERENCE: 3.3.4/Q.774

TITLE: Syntactically invalid behaviour; Invalid structure

SUBTITLE: Subsequent Continue Message; Unknown information element present

PURPOSE: To verify that on receipt of a corrupted Continue message with OTID derivable and DTID derivable and assigned, after transaction establishment, SP A behaves correctly

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) |
|---|---|---|
| | ←———————————— | **BEGIN** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE** | ————————————→ | |
| *Detect error* | ←———————————— | **CONTINUE** |
| **ABORT (P)** | ————————————→ | |
| *TR-P-ABORT ind.*<br><============ | | |

NOTE – If the ABORT message and primitive are not sent, this may be valid behaviour depending 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 CONTINUE MESSAGE CORRECTLY SENT FROM SP A? |
| 6. | CHECK C: IF AN ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH CORRECT DTID AND CORRECT P-ABORT CAUSE VALUE? |
| 7. | CHECK D: IF THE ABORT WAS SENT, WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN IDLE STATE AT SP A? |

| 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 behaviour; 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? |
| 6. | 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

END

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 behaviour; 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.*<br>===========> | | |
| **BEGIN** | ————————→ | |
| *Detect syntax error* | ←———————— | **ABORT** |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message to SP B. |
| 2. | Arrange for SP B to send a corrupted Abort 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 MESSAGE AT SP A. |
| 5. | CHECK C: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE CORRUPTED ABORT MESSAGE. |
| 6. | 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: e.g. incorrect transaction portion 00000011

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.7 1) | | Sheet: 1 of 1 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; 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:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *Detect Unknown message type* | $\longleftarrow$ | **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 (e.g. 01100110)
Message type length: correct number of octets

OTID absent

| | | |
|---|---|---|
| TEST NUMBER: 1.2.2.7 2) | | Sheet: 1 of 1 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Unknown Message; OTID included and DTID not included | | |
| PURPOSE: To verify that on receipt of an Unknown message, 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 an Unknown message with an OTID that is derivable and a DTID that is not derivable or 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) |
|---|---|---|
| *Detect Unknown message type* | ← | **UNKNOWN MESSAGE** |
| **ABORT (P)** | → | |

NOTE – If the Abort message is not sent, this may be valid behaviour depending on the implementation.

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send the Unknown message to SP A. |
| 2. | CHECK A: IF A P-ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH THE CORRECT DTID AND CORRECT P-ABORT CAUSE VALUE? |
| 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? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

UNKNOWN MESSAGE

Message type tag: unknown (e.g. 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)

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: 1.2.2.7 3) | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | |
| TITLE: Syntactically invalid behaviour; Invalid structure | |
| SUBTITLE: Unknown Message; OTID included and DTID included | |
| PURPOSE: To verify that on receipt of an Unknown message with assigned DTID, SP 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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.*<br>============> | | |
| **BEGIN** | ⟶ | |
| *Detect Unknown message type* | ⟵ | **UNKNOWN MESSAGE** |
| **ABORT (P)** | ⟶ | |
| *TR-P-ABORT ind.*<br><============ | | |

NOTE – If the ABORT message and primitive are not sent, this may be valid behaviour 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. | CHECK D: | IF THE ABORT WAS SENT, WERE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE STATE AT SP A? |

| 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 (e.g. 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 behaviour; Invalid encoding

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

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

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 behaviour depending on the implementation.

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send the corrupted Begin message to SP A. |
| 2. | CHECK A: CHECK THAT THE USER WAS NOT INFORMED OF THE BEGIN MESSAGE. |
| 3. | CHECK B: WERE THE TSL STATE MACHINES ASSOCIATED WITH THIS TRANSACTION LEFT IN THE IDLE 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: e.g. 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 |
| REFERENCE: 3.3.4/Q.774 | |
| TITLE: Syntactically invalid behaviour; Invalid encoding | |
| SUBTITLE: Continue Message type; Invalid tag | |
| PURPOSE: To verify that on receipt of a corrupted Continue 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. 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 |

EXPECTED MESSAGE SEQUENCE:

| SP A (TSL) | | SP B (TSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN** | ——————————→ | |
| *Detect syntax error* | ←—————————— | **CONTINUE** |
| **ABORT (P)** | ——————————→ | |
| *TR-P-ABORT ind.*<br><=========== | | |

NOTE – If the ABORT message and the primitive are not sent, this may be valid behaviour 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. | CHECK C: IF AN ABORT MESSAGE WAS SENT, WAS IT SENT CORRECTLY FROM SP A WITH CORRECT DTID AND CORRECT P-ABORT CAUSE VALUE? |

| TEST NUMBER: 1.2.3.2 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Invalid tag: e.g. 00011111<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in BEGIN message)<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: correct number of octets<br>P-Abort cause value: incorrect transaction portion 00000011 | |

| | | |
|---|---|---|
| TEST NUMBER: 1.3.1 1) | | Sheet: 1 of 2 |
| REFERENCE: 3.3.4/Q.774 | | |
| TITLE: Inopportune Messages; Continue message type | | |
| SUBTITLE: Receipt of Continue Message in idle state with unassigned DTID | | |
| PURPOSE: To verify that on receipt of a Continue message with unassigned DTID, signalling point A is able to discard the message and generate an Abort 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 a Continue message with an OTID that is derivable and 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) |
| | $\longleftarrow$ | **CONTINUE** |
| **ABORT (P)** | $\longrightarrow$ | |

TEST DESCRIPTION

| | |
|---|---|
| 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<br><br>Message type tag: 01100101<br>Message type length: correct number of octets<br><br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long)<br><br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br><br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br><br>ABORT (P)<br><br>Message type tag: 01100111<br>Message type length: correct number of octets<br><br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long)<br>(OTID value received in CONTINUE message)<br><br>P-Abort cause tag: 01001010<br>P-Abort cause length: one octet<br>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: Inopportune 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: Inopportune 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 | |

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

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. |
| 2. | CHECK A: VERIFY THAT THE TR-USER WAS NOT INFORMED OF THE ABORT MESSAGE AT SP A. |
| 3. | CHECK B: VERIFY THAT NO MESSAGES WERE GENERATED BY SP A IN RESPONSE TO THE ABORT MESSAGE. |
| 4. | 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 2 |
| REFERENCE: 3.3.3.2/Q.774 | | |
| TITLE: Multiple Transaction Encoding; Valid Transaction Encoding | | |
| SUBTITLE: New transaction request during transaction establishment | | |
| PURPOSE: To verify that the signalling point A is able to correctly react to a Begin message during the establishment of another transaction | | |
| 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.*<br>============> | | |
| **BEGIN** | ⟶ | |
| | ⟵ | **BEGIN** (new transaction) |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-END req.*<br>============> | | |
| *(Basic)*<br>(end new transaction) | | |
| **END** | ⟶ | |
| | ⟵ | **END** |
| *TR-END ind.*<br><============ | | |

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 BY SP A? |
| 9. | CHECK E: WERE TSL STATE MACHINES ASSOCIATED WITH THESE TRANSACTIONS LEFT IN THE IDLE STATE AT SP A? |

| TEST NUMBER: 1.4.1 1) | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN (1st)<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long) X<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>BEGIN (2nd)<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long) Y<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END (1st)<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) Y<br>(OTID value received in 2nd BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>END (2nd)<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) X<br>(OTID value received in 1st BEGIN message)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 1.4.1 2) | | Sheet: 1 of 3 |
| REFERENCE: 3.3.3.2/Q.774 | | |
| TITLE: Multiple Transaction Encoding; Valid Transaction Encoding | | |
| SUBTITLE: New transaction request after transaction establishment | | |
| PURPOSE: To verify that the signalling point A is able to correctly react to a Begin message after the establishment of another transaction | | |
| 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.*<br>============> | | |
| **BEGIN** | ⟶ | |
| | ⟵ | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |
| | ⟵ | **BEGIN** (new transaction) |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-END req.*<br>============> | | |
| *(Basic)*<br>(end new transaction) | | |
| **END** | ⟶ | |
| | ⟵ | **END** |
| *TR-END ind.*<br><============ | | |

| 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. | |
| 6. | CHECK A: WAS THE FIRST BEGIN MESSAGE CORRECTLY SENT BY SP A? | |
| 7. | CHECK B: WAS THE CONTINUE MESSAGE CORRECTLY RECEIVED BY SP A? | |
| 8. | CHECK C: WAS THE SECOND BEGIN MESSAGE CORRECTLY RECEIVED BY SP A? | |
| 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)<br><br>Message type tag: 01100010<br>Message type length: correct number of octets<br><br>Destination transaction ID tag: 01001000<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) X<br><br>Component portion tag: 01101100<br>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 2 |
| REFERENCE: 3.3.3.2/Q.774 | | |
| TITLE: Multiple Transaction Encoding; Inopportune Messages | | |
| SUBTITLE: Message with unassigned DTID during transaction establishment | | |
| PURPOSE: To verify that the signalling point A is able to correctly react to a Continue message with DTID unassigned during the establishment of another transaction | | |
| 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.* <============ | | |
| | ⟵ | **END** |
| *TR-END ind.* <============ | | |

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? |
| 7. | CHECK D: WAS THE P-ABORT CAUSE IN THE ABORT MESSAGE THE CORRECT VALUE, (UNRECOGNIZED TRANSACTION ID)? |
| 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: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| BEGIN<br>Message type tag: 01100010<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long) X<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>CONTINUE<br>Message type tag: 01100101<br>Message type length: correct number of octets<br>Originating transaction ID tag: 01001000<br>Originating transaction ID length: correct number of octets<br>Originating transaction ID value: OCTET STRING (1-4 octets long) Y<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) Z (Not equal to X)<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>ABORT (P)<br>Message type tag: 01100111<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) Y (OTID value received in CONTINUE message)<br>P-Abort cause tag: 01001010<br>P-Abort cause length: one octet<br>P-Abort cause value: 00000001 Unrecognized Transaction ID<br>END<br>Message type tag: 01100100<br>Message type length: correct number of octets<br>Destination transaction ID tag: 01001001<br>Destination transaction ID length: correct number of octets<br>Destination transaction ID value: OCTET STRING (1-4 octets long) X (OTID value received in BEGIN message)<br>Component portion tag: 01101100<br>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 DTID after transaction establishment | | |
| PURPOSE: To verify that the signalling point A is able to correctly react to a Continue message with DTID unassigned after the establishment of another transaction | | |
| 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.*<br>============> | | |
| **BEGIN** | ————————————→ | |
| | ←———————————— | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |
| | ←———————————— | **CONTINUE** (new transaction) |
| **ABORT (P)** | ————————————→ | |
| *TR-P-ABORT ind.*<br><============ | | |
| | ←———————————— | **END** |
| *TR-END ind.*<br><============ | | |

| 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. |
| 5. | CHECK A: WAS THE BEGIN MESSAGE CORRECTLY SENT BY SP A? |
| 6. | CHECK B: WERE THE CONTINUE MESSAGES CORRECTLY RECEIVED BY SP A? |
| 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? |
| 10. | 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

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: 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 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 (CSL) 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.

g) For the dialogue portion tests, the check table sometimes has a "protocol version" shown and sometimes it is not shown. In the main recommendations this information is optional and if not present the default value is "version 1".

### 7.2.2 Component sublayer test list

All tests are validation tests

Tests marked "*" are compatibility tests

*2 Component Sublayer*

2.1 Valid Functions

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

2.1.2 Invoke component, linked operations

2.1.2.1 Class 1 original 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 Class 4 original operation invocation

* 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 Reject

2.1.3.1 Remote Reject by CSL

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 Remote Reject 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 Remote Reject 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 component leading to TC-User reject

2.1.4.1 Invoke problem

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 Result problem

2.1.4.2.1 Wrong type parameter

2.1.4.3 Return Error 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 Segmentation for Return Result

2.1.5.1 Class 1 single 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 Class 3 single operation invocation

* 2.1.5.2.1 IUT as sender: receive segmented components

2.1.6 User Cancel

2.1.7 Encoding variations

2.1.7.1 Component length definite short

2.1.7.2 Component length definite long

2.1.7.3 Component length indefinite

2.1.7.4 Value variations

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 Multiple components grouping

2.1.8.1 Multiple operations invocation; receiving success

2.1.8.2 Multiple operations invocation; reporting success

2.1.8.3 A malformed component received

2.1.9 Dialogue Portion

2.1.9.1 Accept application context proposal

2.1.9.1.1 Send AARQ in begin message

2.1.9.1.2 Accept AARQ and continue dialogue

2.1.9.1.3 Accept AARQ and end dialogue

2.1.9.2 Propose alternative Application context

2.1.9.2.1 Send AARE with alternative

2.1.9.2.2 Receive AARE with alternative

2.1.9.3 Dialogue refused

2.1.9.4 Dialogue abort

2.1.9.5 Transport of User Information

2.1.9.5.1 Accept User Information in Begin message

2.1.9.5.2 Accept User Information in first Continue message

2.1.9.5.3 Accept User Information in subsequent Continue message

2.1.9.5.4 Accept several User Information elements in Continue message

2.1.9.6 Unstructured dialogue

2.1.9.7 Dialogue control APDU Version

2.1.9.7.1 Structured dialogue, NOT Version 1

2.1.9.7.2 Structured dialogue, Version 1

2.1.9.7.3 Unstructured dialogue, NOT Version 1

2.1.9.7.4 Unstructured dialogue, Version 1

2.2 Syntactically invalid behaviour

2.2.1 Invalid values for information elements

2.2.1.1 Length of Invoke ID > 1 in Invoke component

2.2.1.2 Length of Invoke ID = 0 in Invoke component

2.2.2 Invalid structure

2.2.2.1 Invoke component

2.2.2.1.1 Invoke ID missing

2.2.2.1.2 Operation code missing

2.2.2.2 Return Result component

2.2.2.2.1 Invoke ID missing

2.2.2.2.2 Operation code missing while parameters included

2.2.2.2.3 Sequence tag missing while parameters included

- 2.2.2.3 Return Error
  - 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
- 2.2.2.5 Dialogue Portion
  - 2.2.2.5.1 Missing application context in APDU AARQ
  - 2.2.2.5.2 Incorrect length
  - 2.2.2.5.3 Missing result-source-diagnostic
  - 2.2.2.5.4 Missing application context in APDU AUDT
  - 2.2.2.5.5 External type without direct reference
  - 2.2.2.5.6 Indirect reference in external type
  - 2.2.2.5.7 User information without direct reference
  - 2.2.2.5.8 Indirect reference in User information

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 Inopportune behaviour
- 2.3.1 Inopportune Invoke component
  - 2.3.1.1 Invalid linked ID
- 2.3.2 Unrecognized Invoke ID
  - 2.3.2.1 Inopportune Return Result-Last component
  - 2.3.2.2 Inopportune Return Result Not-Last component
  - 2.3.2.3 Inopportune Return Error component
  - 2.3.2.4 Inopportune Reject component
- 2.3.3 Unexpected Components
  - 2.3.3.1 Return Result-Last for Class 2
  - 2.3.3.2 Return Result-Last for Class 4
  - 2.3.3.3 Return Result Not-Last for Class 2
  - 2.3.3.4 Return Result Not-Last for Class 4
  - 2.3.3.5 Return Error for Class 3
  - 2.3.3.6 Return Error for Class 4
- 2.3.4 Dialogue Portion, unexpected APDUs
  - 2.3.4.1 Begin message with APDU AARE
  - 2.3.4.2 Dialogue confirmation with any APDU other than AARE
  - 2.3.4.3 Dialogue confirmation with APDU ABRT
  - 2.3.4.4 Dialogue continuation with the presence of a Dialogue PDU in the active state
  - 2.3.4.5 Unidirectional message with unexpected abstract syntax
  - 2.3.4.6 Unexpected dialogue portion in Continue message
  - 2.3.4.7 Missing dialogue portion in Continue message
  - 2.3.4.8 Begin message with unexpected abstract syntax

| | |
|---|---|
| TEST NUMBER: 2.1.1.1.1 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774

TITLE: Valid functions; Invoke component, unlinked operations

SUBTITLE: Class 1 single operation invocation; IUT as sender: receive result

PURPOSE: To verify that a single Class 1 operation can be successfully invoked and the successful completion of the operation 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 Return Result-Last 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)** | $\longrightarrow$ | |
| | $\longleftarrow$ | **RETURN RESULT-LAST (i)** |
| *TC-RESULT-L ind.* <============ | | |

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 RETURN RESULT-LAST 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

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 NUMBER: 2.1.1.1.2 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Valid functions; Invoke component, unlinked operations | | |
| SUBTITLE: Class 1 single operation invocation; IUT as receiver: report result | | |
| PURPOSE: To verify that a Class 1 operation can be successfully invoked and the successful completion of the operation can be sent | | |
| PRE-TEST CONDITIONS: Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B 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)

$\longleftarrow$ **INVOKE (i)**

*TC-INVOKE ind.*
<============

*TC-RESULT-L req.*
============>

**RETURN-RESULT-LAST (i)** $\longrightarrow$

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single 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 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

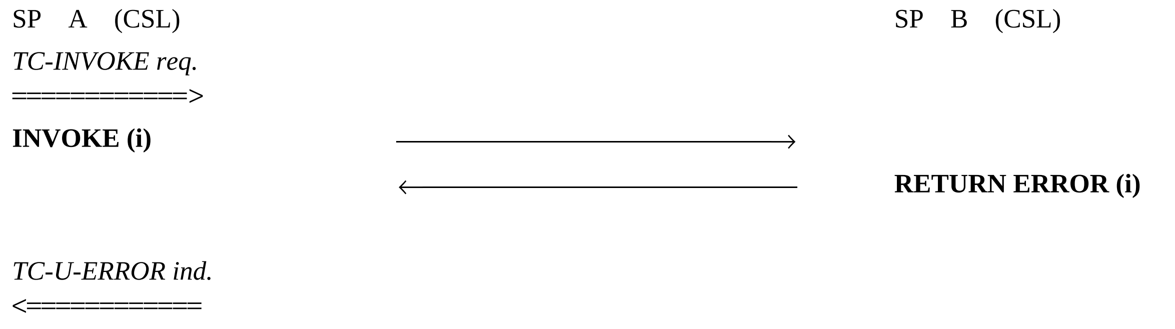
Component type tag: 10100001 (INVOKE)
Component length: correct number of octets

Invoke ID tag: 00000010
Invoke ID length: 00000001 (one octet)
Invoke ID: 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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in TSL messages from SP A to SP B<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br><br>parameters (provided by the TC-User)<br><br>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 component, unlinked operations | | |
| SUBTITLE: Class 1 single operation invocation; IUT as sender: receive error | | |
| PURPOSE: To verify that a Class 1 operation can be successfully invoked and the unsuccessful completion of the operation can be received and delivered to the TC-User | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>2) Arrange the data at SP B such that a Return Error component can be generated | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT and CPT | TYPE OF SP: SP |

EXPECTED MESSAGE AND COMPONENT FLOW:



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 RETURN ERROR 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

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN ERROR component in TSL messages from SP B to SP A<br><br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y (y is a valid error code)<br><br>parameters (provided by the TC-User) | |

| | |
|---|---|
| TEST NUMBER: 2.1.1.1.4 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774

TITLE: Valid functions; Invoke component, unlinked operations

SUBTITLE: Class 1 single operation invocation; IUT as receiver: report error

PURPOSE: To verify that a Class 1 operation can be successfully invoked and the unsuccessful completion of the operation can be sent

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 at SP A such that a Return-Error 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) |
|---|---|---|
| | ← | **INVOKE (i)** |
| *TC-U-ERROR ind.*<br><============ | | |
| *TC-RESULT-L req.*<br>============> | | |
| **RETURN-ERROR (i)** | → | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single 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 THE RETURN ERROR COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 4. | CHECK C: WAS THE INVOKE ID IN THE RETURN ERROR COMPONENT THE SAME AS THE ONE IN THE INVOKE COMPONENT? |
| 5. | CHECK D: WAS THE ERROR CODE IN THE RETURN ERROR COMPONENT 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

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: 2.1.1.1.4 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN ERROR COMPONENT in TSL message from SP A to SP B<br><br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y<br><br>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:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>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)** $\longrightarrow$

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 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Valid functions; Invoke component, unlinked operations | | |
| SUBTITLE: Class 2 single operation invocation; IUT as sender: receive error | | |
| PURPOSE: To verify that a Class 2 operation can be successfully invoked and the failure report can be received and delivered to the TC-User | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>2) Arrange the data at SP B such that a Return Error 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.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **RETURN ERROR (i)** |
| *TC-U-ERROR ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single Class 2 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 ERROR 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

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN ERROR COMPONENT in TSL message from SP B to SP A<br><br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y<br><br>parameters (provided by the TC-User) | |

| | |
|---|---|
| TEST NUMBER: 2.1.1.2.2 | Sheet: 1 of 1 |

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.*<br>============> | | |
| **INVOKE (i)**<br>timer expiry for invocation (i) | $\longrightarrow$ | |
| *TC-L-CANCEL ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single Class 2 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 TC-USER AT SP A INFORMED OF TIMER EXPIRY? |
| 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 |

REFERENCE: 3.2.1/Q.774

TITLE: Valid functions; Invoke component, unlinked operations

SUBTITLE: Class 3 single operation invocation; 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-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 and CPT | TYPE OF SP: SP |

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>===========> | | |
| **INVOKE (i)** | ——————————————→ | |
| | ←—————————————— | **RETURN RESULT-LAST (i)** |
| *TC-RESULT-L ind.*<br><=========== | | |

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. | 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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br>Component type tag: 10100011 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>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 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.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| timer expiry for invocation (i) | | |
| *TC-L-CANCEL ind.*<br><============ | | |

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? |
| 3. | CHECK B: WAS THE COMPONENT FLOW AS SHOWN ABOVE? |
| 4. | CHECK C: WAS THE TC-USER AT SP A INFORMED OF TIMER EXPIRY? |
| 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: 2.1.1.4.1 | | 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 |

REFERENCE: 3.2.1/Q.774

TITLE: Valid functions; Invoke component, Linked operations

SUBTITLE: Class 1 original operation invocation; IUT as sender: receive a linked Class 1 operation invocation, report result

PURPOSE: To verify that a linked Class 1 operation can be successfully received and the successful completion of the original operation can be performed

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 linked Invoke 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.*<br>============> | | |
| **INVOKE (i)** | ——————————→ | |
| | ←—————————— | **INVOKE (j, i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-RESULT-L req.*<br>============> | | |
| **RETURN-RESULT-LAST (j)** | ——————————→ | |
| | ←—————————— | **RETURN-RESULT-LAST (i)** |
| *TC-RESULT-L ind.*<br><============ | | |

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: 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: 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<br><br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br><br>INVOKE component in TSL message from SP A to SP B<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>INVOKE component in TSL message sent by SP B<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: j (j represents an integer)<br><br>Linked ID tag: 10000000<br>Linked ID length: 00000001 (one octet)<br>Linked ID: i<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Operation code: y (y represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in 2nd TSL message sent by SP A<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>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)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long) (see Note)<br>Operation code: y (see Note)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br><br>parameters (provided by the TC-User)<br><br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.2.1.2 | | Sheet: 1 of 3 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Valid functions; Invoke component, Linked operations | | |
| SUBTITLE: Class 1 original operation invocation; IUT as receiver: send a linked Class 1 operation invocation, receive result | | |
| PURPOSE: To verify that a linked Class 1 operation can be successfully invoked and the successful completion of the original operation can be performed | | |
| PRE-TEST CONDITIONS: Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains an Invoke component which will invoke a linked operation | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT and CPT | TYPE OF SP: SP |

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ⟵ | **INVOKE (i)** |
| *TC-INVOKE ind.* <============ | | |
| *TC-INVOKE req.* ============> | | |
| **INVOKE (j, i)** | ⟶ | |
| | ⟵ | **RETURN RESULT-LAST (j)** |
| *TC-RESULT-L ind.* <============ | | |
| *TC-RESULT-L req.* ============> | | |
| **RETURN-RESULT-LAST (i)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | 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 INVOKE COMPONENT WITH CORRECT INFORMATION 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<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>INVOKE component in initial TSL message from SP B to SP A<br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>INVOKE component in TSL message sent from SP A to SP B<br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: j (j represents an integer)<br>Linked ID tag: 10000000<br>Linked ID length: 00000001 (one octet)<br>Linked ID: i<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Operation code: y (y represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message sent by SP B<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>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)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long) (see Note)<br>Operation code: y (see Note)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message sent by SP A<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.2.1.3 | | Sheet: 1 of 3 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Valid functions; Invoke component, Linked operations | | |
| SUBTITLE: Class 1 original operation invocation; IUT as sender: receive a linked Class 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 CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>2) Arrange the data at SP B such that a linked invocation 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.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **INVOKE (j, i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-U-ERROR req.*<br>============> | | |
| **RETURN-ERROR (j)** | ⟶ | |
| | ⟵ | **RETURN-RESULT-LAST (i)** |
| *TC-RESULT-L ind.*<br><============ | | |

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: WAS A LINKED INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A? |
| 4. | CHECK C: WAS THE RETURN ERROR COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 5. | 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<br>    Component portion tag: 01101100<br>    Component portion length: correct number of octets<br>INVOKE component in TSL message from SP A to SP B<br>    Component type tag:  10100001 (INVOKE)<br>    Component length:  correct number of octets<br>    Invoke ID tag:  00000010<br>    Invoke ID length:  00000001 (one octet)<br>    Invoke ID:   i (i represents an integer)<br>    Operation code tag:  00000010 (local) or 00000110 (global)<br>    Operation code length:  correct number of octets (e.g. 00000001 if x is one octet long)<br>    Operation code:  x (x represents a valid operation code)<br>    parameters (provided by the TC-User)<br>INVOKE component in the TSL messages sent by SP B<br>    Component type tag:  10100001 (INVOKE)<br>    Component length:  correct number of octets<br>    Invoke ID tag:  00000010<br>    Invoke ID length:  00000001 (one octet)<br>    Invoke ID:   j (j represents an integer)<br>    Linked ID tag:  10000000<br>    Linked ID length:  00000001 (one octet)<br>    Linked ID:  i<br>    Operation code tag:  00000010 (local) or 00000110 (global)<br>    Operation code length:  correct number of octets (e.g. 00000001 if y is one octet long)<br>    Operation code:  y (y represents a valid operation code)<br>    parameters (provided by the TC-User)<br>RETURN ERROR component in the TSL message sent by SP A<br>    Component type tag:  10100010 (RETURN ERROR)<br>    Component length:  correct number of octets<br>    Invoke ID tag:  00000010<br>    Invoke ID length:  00000001<br>    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)<br>Error code length: correct number of octets (e.g. 00000001 if z is one octet long) (see Note)<br>Error code: z (see Note)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in the TSL message sent by SP B<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br><br>parameters (provided by the TC-User)<br><br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.2.1.4 | | Sheet: 1 of 3 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Valid functions; Invoke component, Linked operations | | |
| SUBTITLE: Class 1 original operation invocation; IUT as receiver: send a linked Class 1 operation invocation, receive error | | |
| PURPOSE: To verify that a linked Class 1 operation can be successfully invoked and the receiving error will not impact the completion of the original operation | | |
| PRE-TEST CONDITIONS: Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B 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) |
|---|---|---|
| | ←―――――――――――― | **INVOKE (i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (j, i)** | ――――――――――――→ | |
| | ←―――――――――――― | **RETURN ERROR (j)** |
| *TC-U-ERROR ind.*<br><============ | | |
| *TC-RESULT-L req.*<br>============> | | |
| **RETURN-RESULT-LAST (i)** | ――――――――――――→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | 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 INVOKE COMPONENT WITH CORRECT INFORMATION 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 ERROR COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A? |
| 6. | CHECK E: WAS THE RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 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<br><br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br><br>INVOKE component in TSL message from SP B to SP A<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>INVOKE component in TSL messages by SP A<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: j (j represents an integer)<br><br>Linked ID tag: 10000000<br>Linked ID length: 00000001 (one octet)<br>Linked ID: i<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Operation code: y (y represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN ERROR component in TSL message sent by SP B<br><br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>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)<br>Error code length: correct number of octets (e.g. 00000001 if z is one octet long)<br>Error code: z<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in TSL message sent by SP A<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br><br>parameters (provided by the TC-User)<br><br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.2.2.1 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774

TITLE: Valid functions; Invoke component, Linked operations

SUBTITLE: Class 4 original operation invocation; IUT as sender: receive a linked Class 2 operation invocation, no outcome

PURPOSE: To verify that a linked Class 2 operation can be successfully received and the successful completion of the original Class 4 operation can be performed

PRE-TEST CONDITIONS:

1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains a Class 4 Invoke component
2) Arrange the data at SP B such that a linked Class 2 Invoke 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.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **INVOKE (j, i)** |
| *TC-INVOKE ind.*<br><============ | | |
| timer expiry for invocation (i) | | |
| *TC-L-CANCEL ind.*<br><============ | | |

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: WAS A LINKED INVOKE 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

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>INVOKE component in TSL message sent by SP B<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: j (j represents an integer)<br><br>Linked ID tag: 10000000<br>Linked ID length: 00000001 (one octet)<br>Linked ID: i<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Operation code: y (y represents a valid operation code)<br><br>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 component, 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 Class 2 operation can be successfully invoked and the successful completion of the original Class 4 operation can be performed | |
| PRE-TEST CONDITIONS: Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains an Invoke component which will invoke a Class 2 linked operation | |

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

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←――――――――――― | **INVOKE (i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (j, i)** | ―――――――――――→ | |
| timer expiry for invocation (j) | | |
| *TC-L-CANCEL ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | 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 INVOKE COMPONENT WITH CORRECT INFORMATION 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 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 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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>INVOKE component in TSL message sent from SP A to SP B<br><br>Component type tag: 10100010 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: j (j represents)<br><br>Linked ID tag: 10000000<br>Linked ID length: 00000001 (one octet)<br>Linked ID: i<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Operation code: y (y represents a valid operation code)<br><br>parameters (provided by the TC-User) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.3.1.1 | | Sheet: 1 of 2 |
| REFERENCES: 3.2.1/Q.774; 3.7.1/Q.772 | | |
| TITLE: Valid functions; Remote Reject | | |
| SUBTITLE: Remote Reject by CSL; General problem code | | |
| PURPOSE: To verify that a remote rejection by CSL with general problem code can be delivered to the TC-User | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>2) Arrange the data at SP B such that a Reject component with general problem code can be generated | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE AND COMPONENT FLOW:

$$\begin{tikzcd}
\text{SP A (CSL)} & & \text{SP B (CSL)} \\
\textit{TC-INVOKE req.} \; \text{============>} & & \\
\textbf{INVOKE (i)} \arrow[rr] & & {} \\
{} & & \textbf{REJECT (i)} \arrow[ll] \\
\textit{TC-R-REJECT ind.} \; \text{<============} & &
\end{tikzcd}$$

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single Class 1 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 REJECT COMPONENT WITH CORRECT INFORMATION PAS SED 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

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message sent from SP B to SP A<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000000 (General Problem)<br>Problem code length: 00000001<br>Problem code: 00000000 (unrecognized component) | |

| | |
|---|---|
| TEST NUMBER: 2.1.3.1.2 | Sheet: 1 of 2 |

REFERENCES: 3.2.1/Q.774; 3.7.2/Q.772

TITLE: Valid functions; Remote Reject

SUBTITLE: Remote Reject by CSL; Invoke problem code

PURPOSE: To verify that the remote rejection by CSL with Invoke problem code 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 B contains an Invoke component
2) Arrange the data at SP B such that a Reject component with Invoke problem code can be generated

| | | |
|---|---|---|
| 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-INVOKE req.* ============> | | |
| **INVOKE (j,i)** | ⟶ | |
| | ⟵ | **REJECT (j)** |
| *TC-R-REJECT ind.* <============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a linked Class 1 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 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 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; 3.7.3/Q.772

TITLE: Valid functions; Remote Reject

SUBTITLE: Remote Reject by CSL; Return Result problem code

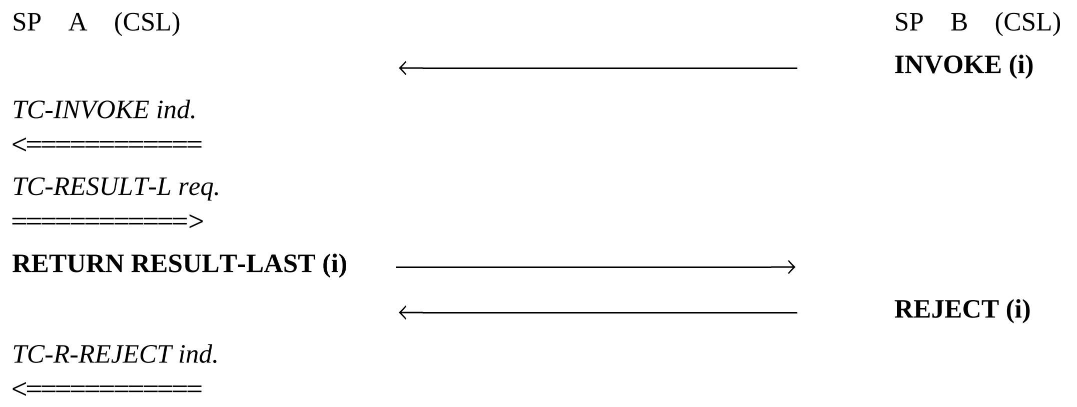
PURPOSE: To verify that a single Class 1 operation can be successfully invoked and the remote rejection can be received and delivered to the TC-User

PRE-TEST CONDITIONS:

1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B 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 COMPONENT FLOW:



TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single Class 1 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 THE REJECT COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER 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: 2.1.3.1.3 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in TSL message from SP A to SP B<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br><br>Parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP B to SP A<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000010 (RETURN RESULT)<br>Problem code length: 00000001<br>Problem code: 00000000 (unrecognized Invoke ID)<br><br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.3.1.4 | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774; 3.7.4/Q.772 | |
| TITLE: Valid functions; Remote Reject | |
| SUBTITLE: Remote Reject by CSL; Return Error problem code | |
| PURPOSE: To verify that a single Class 1 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 B 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←―――――――――――――― | **INVOKE (i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-U-ERROR req.*<br>============> | | |
| **RETURN ERROR (i)** | ――――――――――――――→ | |
| | ←―――――――――――――― | **REJECT (i)** |
| *TC-R-REJECT ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single Class 1 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 THE REJECT COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER 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: 2.1.3.1.4 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN ERROR component in TSL message from SP A to SP B<br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP B to SP A<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000011 (RETURN ERROR)<br>Problem code length: 00000001<br>Problem code: 00000001 (unrecognized Invoke ID) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.3.2.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774; 3.7.2/Q.772 | | |
| TITLE: Valid functions; Remote Reject | | |
| SUBTITLE: Remote Reject by TC-User; Invoke problem code | | |
| PURPOSE: To verify that the remote rejection by TC-User with Invoked problem code can be received and delivered to the TC-User | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | $\longrightarrow$ | |
| | $\longleftarrow$ | **REJECT (i)** |
| *TC-U-REJECT ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single Class 1 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 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

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents an operation code)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP B to SP A<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000001 (RETURN ERROR)<br>Problem code length: 00000001<br>Problem code: 00000000 (unrecognized Invoke ID) | |

| | |
|---|---|
| TEST NUMBER: 2.1.3.2.2 | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774; 3.7.3/Q.772 | |
| TITLE: Valid functions; Remote Reject | |
| SUBTITLE: Remote Reject by TC-User; Return Result problem code | |
| PURPOSE: To verify that the remote rejection by TC-User with Return Result problem code 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 B contains an Invoke component
2) Arrange the data at SP B such that a Reject component with Return Result problem code can be generated

| | | |
|---|---|---|
| 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)** ————————————→

←———————————— **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 INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A? |
| 3. | CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION PAS SED TO TC-USER 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: 2.1.3.2.2 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents an operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP A to SP B<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP B to SP A<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000010 (RETURN RESULT)<br>Problem code length: 00000001<br>Problem code: 00000010 (wrong type parameter)<br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.3.2.3 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774; 3.7.4/Q.772

TITLE: Valid functions; Remote Reject

SUBTITLE: Remote Reject by TC-User; Return Error problem code

PURPOSE: To verify that the remote rejection by TC-User with Return Error problem code 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 B contains an Invoke component
2) Arrange the data at SP B such that a Reject component with Return Error problem code can be generated

| | | |
|---|---|---|
| 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-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 INVOKE COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A?
3. CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER 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: 2.1.3.2.3 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents an operation code)<br>parameters (provided by the TC-User)<br>RETURN ERROR component in TSL message from SP A to SP B<br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP B to SP A<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000011 (RETURN ERROR)<br>Problem code length: 00000001<br>Problem code: 00000010 (unrecognized error) | |

| | |
|---|---|
| TEST NUMBER: 2.1.3.3.1 | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774 | |
| TITLE: Valid functions; Remote Reject | |
| SUBTITLE: Remote Reject with an Invoke problem code; Class 1 operation invocation | |
| PURPOSE: To verify that a single Class 1 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.* ============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **REJECT (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 INVOKE COMPONENT WITH CORRECT INFORMATION 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 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: 2.1.3.3.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP B to SP A<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000001 (INVOKE)<br>Problem code length: 00000001<br>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

SUBTITLE: Remote Reject with an Invoke problem code; Class 2 operation invocation

PURPOSE: To verify that a single Class 2 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | $\longrightarrow$ | |
| | $\longleftarrow$ | **REJECT (i)** |
| *TC-U-REJECT ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single Class 2 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 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

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP B to SP A<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000001 (INVOKE)<br>Problem code length: 00000001<br>Problem code: 00000010 (wrong type parameter) | |

| | |
|---|---|
| TEST NUMBER: 2.1.3.3.3 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774

TITLE: Valid functions; Remote Reject

SUBTITLE: Remote Reject with an Invoke problem code; Class 3 operation invocation

PURPOSE: To verify that a single Class 3 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.* ============> | | |
| **INVOKE (i)** | $\longrightarrow$ | |
| | $\longleftarrow$ | **REJECT (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 INVOKE COMPONENT WITH CORRECT INFORMATION 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

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents an operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP B to SP A<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000001 (INVOKE)<br>Problem code length: 00000001<br>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 | |
| SUBTITLE: Remote Reject with an Invoke 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | $\longrightarrow$ | |
| | $\longleftarrow$ | **REJECT (i)** |
| *TC-U-REJECT ind.*<br><============ | | |

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 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

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP B to SP A<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000001 (INVOKE)<br>Problem code length: 00000001<br>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 leading to TC-User reject | | |
| SUBTITLE: Invoke problem; Unrecognized operation code | | |
| PURPOSE: To verify that a rejection of a requested operation 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:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←―――――――――――――― | **INVOKE (i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-U-REJECT req.*<br>============> | | |
| **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 COMPONENT WITH CORRECT INFORMATION 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 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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents an invalid operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message sent from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000001 (INVOKE problem type)<br>Problem code length: 00000001<br>Problem code: 00000001 (unrecognized operation) | |

| | |
|---|---|
| TEST NUMBER: 2.1.4.1.2 | Sheet: 1 of 3 |

REFERENCE: 3.2.2/Q.774

TITLE: Valid functions; Reception of component leading to TC-User reject

SUBTITLE: Invoke problem; Unexpected linked operation

PURPOSE: To verify that a rejection can be successfully initiated due to an unexpected linked operation and without affecting the original invocation.

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 Invoke with a linked ID is contained in an appropriate TSL message

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

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **INVOKE (j, i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-U-REJECT req.*<br>============> | | |
| **REJECT (j)** | ⟶ | |
| | ⟵ | **RETURN RESULT-LAST (i)** |
| *TC-RESULT-L ind.*<br><============ | | |

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 INVOCATION 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<br><br>Component type tag:   10100010 (RETURN RESULT-LAST)<br>Component length:   correct number of octets<br><br>Invoke ID tag:   00000010<br>Invoke ID length:   00000001<br>Invoke ID:   i<br><br>Sequence tag:   00110000 (see Note)<br>Sequence length:   corect number of octets (see Note)<br><br>Operation code tag:   00000010 (local) or 00000110 (global) (see Note)<br>Operation code length:   correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code:   x (see Note)<br><br>parameters (provided by the TC-User)<br><br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.4.1.3 | Sheet: 1 of 3 |
| REFERENCE: 3.2.1/Q.774 | |
| TITLE: Valid functions; Reception of component leading to TC-User reject | |
| SUBTITLE: Invoke problem; Linked response unexpected | |
| PURPOSE: To verify that an unexpected linked response can be rejected | |

PRE-TEST CONDITIONS:

1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains an Invoke component which will invoke a linked operation
2) Arrange the data at SP B such that a linked response contains at least one parameter which is not associated with the outcome of the operation

| | | |
|---|---|---|
| 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)** | ⟶ | |
| | ⟵ | **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 RETURN RESULT-LAST COMPONENT WITH CORRECT INFORMATION PASSED TO TC-USER BY SP A? |
| 6. | CHECK E: WAS THE INVOCATION 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<br><br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br><br>INVOKE component in TSL message from SP A to SP B<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code that does not allow any linked operation)<br><br>parameters (provided by the TC-User)<br><br>INVOKE component in the TSL message sent from SP B to SP A<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: j (j represents an integer)<br><br>Linked ID tag: 10000000<br>Linked ID length: 00000001 (one octet)<br>Linked ID: i<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Operation code: y (y represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message sent by SP A<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>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)<br>Problem code length: 00000001<br>Problem code: 00000111 (linked response unexpected)<br>RETURN RESULT-LAST component in TSL message by SP B<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: corect number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.4.1.4 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.2/Q.774 | | |
| TITLE: Valid functions; Reception of component leading to TC-User reject | | |
| SUBTITLE: Invoke problem; Wrong type parameter | | |
| PURPOSE: To verify that a rejection of a requested operation 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:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | $\longleftarrow$ | **INVOKE (i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-U-REJECT req.*<br>============> | | |
| **REJECT (i)** | $\longrightarrow$ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP B to SP A with a wrong type parameter included. |
| 2. | CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION 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 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: 2.1.4.1.4 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User, including at least one parameter which is not one of those associated with the operation)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000001 (Invoke problem type)<br>Problem code length: 00000001<br>Problem code: 00000010 (wrong type parameter) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.4.2.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2/Q.774 | | |
| TITLE: Valid functions; Reception of component leading to TC-User reject | | |
| SUBTITLE: Return Result problem; Wrong type parameter | | |
| PURPOSE: To verify that a rejection can be successfully initiated due to an invalid operation code included in the Return Result-Last component | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component for Class 1 or 3<br>2) Arrange the data at SP B such that a Return Result-Last with an invalid operation code is generated | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **RETURN RESULT-LAST (i)** |
| *TC-RESULT-L ind.*<br><============ | | |
| *TC-U-REJECT req.*<br>============> | | |
| **REJECT (i)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP A to SP B.<br>Generate a response from SP B to SP A with a valid Invoke ID but a different operation code. |
| 2. | CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 3. | CHECK B: WAS THE RETURN RESULT-LAST COMPONENT PASSED TO TC-USER BY SP A? |
| 4. | CHECK C: WAS THE REJECT COMPONENT SENT BY SP A? |
| 5. | CHECK D: WAS THE INVOCATION 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<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>INVOKE component in TSL message from SP A to SP B<br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br>Operation code tag: 00000010 (local)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if y is one octet long) (see Note)<br>Operation code: y (y is different from x) (see Note)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000010 (RETURN RESULT)<br>Problem code length: 00000001<br>Problem code: 00000010 (wrong type parameter)<br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.4.3.1 | Sheet: 1 of 2 |

REFERENCE: 3.2.2/Q.774

TITLE: Valid functions; Reception of component leading to TC-User reject

SUBTITLE: Return Error problem; Unrecognized error

PURPOSE: To verify that a rejection can be successfully initiated due to an unrecognized error code included in the Return Error component

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 invalid error code is generated

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

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **RETURN ERROR (i)** |
| *TC-U-ERROR ind.*<br><============ | | |
| *TC-U-REJECT req.*<br>============> | | |
| **REJECT (i)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate A Class 1 operation invocation from SP A to SP B.<br>Generate an unsuccessful response from SP B to SP A with a valid Invoke ID but an invalid error code for this operation. |
| 2. | CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 3. | CHECK B: WAS THE RETURN ERROR COMPONENT PASSED TO TC-USER BY SP A? |
| 4. | CHECK C: WAS THE REJECT COMPONENT SENT BY SP A? |
| 5. | CHECK D: WAS THE INVOCATION 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<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>INVOKE component in TSL message from SP A to SP B<br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN ERROR component in TSL message from SP B to SP A<br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y (y is an invalid error code for this operation)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000011 (RETURN ERROR)<br>Problem code length: 00000001<br>Problem code: 00000010 (unrecognized error) | |

| | |
|---|---|
| TEST NUMBER: 2.1.4.3.2 | Sheet: 1 of 2 |

REFERENCE: 3.2.2/Q.774

TITLE: Valid functions; Reception of component leading to TC-User reject

SUBTITLE: Return Error problem; Unexpected error

PURPOSE: To verify that a rejection can be successfully initiated due to an unexpected error code included in the Return Error component

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 |

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | $\longrightarrow$ | |
| | $\longleftarrow$ | **RETURN ERROR (i)** |
| *TC-U-ERROR ind.*<br><============ | | |
| *TC-U-REJECT req.*<br>============> | | |
| **REJECT (i)** | $\longrightarrow$ | |

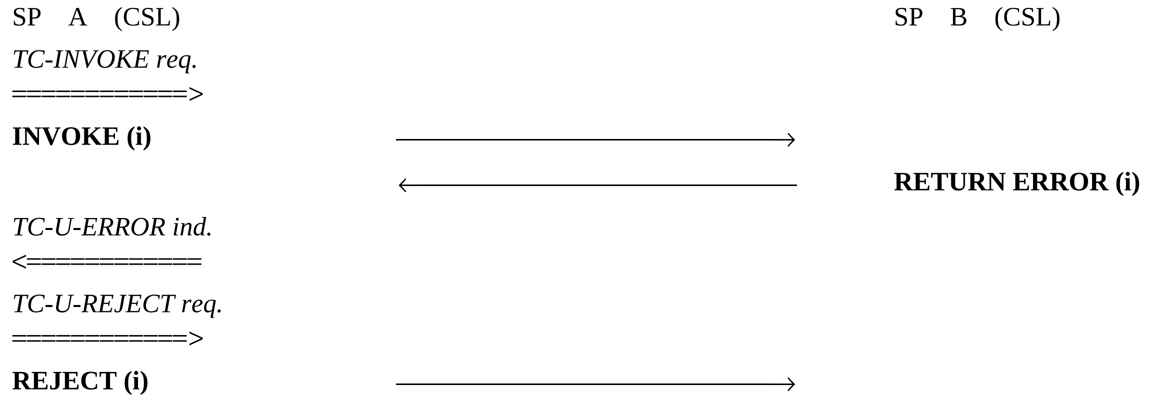
TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a Class 1 operation invocation from SP A to SP B.<br>Generate an unsuccessful response from SP B to SP A with a valid Invoke ID but an unexpected error code for this operation. |
| 2. | CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 3. | CHECK B: WAS THE RETURN ERROR COMPONENT PASSED TO TC-USER BY SP A? |
| 4. | CHECK C: WAS THE REJECT COMPONENT SENT BY SP A? |
| 5. | CHECK D: WAS THE INVOCATION 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<br><br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br><br>INVOKE component in TSL message from SP A to SP B<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN ERROR component in TSL message from SP B to SP A<br><br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y (y is an error code that is not one of those which the invoked operation may report)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000011 (RETURN ERROR)<br>Problem code length: 00000001<br>Problem code: 00000011 (unexpected error) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.4.3.3 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2/Q.774 | | |
| TITLE: Valid functions; Reception of component leading to TC-User reject | | |
| SUBTITLE: Return Error problem; Wrong type parameter | | |
| PURPOSE: To verify that a rejection can be successfully initiated due to a wrong type parameter included in the Return Error component | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component of Class 1<br>2) Arrange the data at SP B such that a Return Error with a wrong type parameter is generated | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE AND COMPONENT FLOW:



| TEST DESCRIPTION | |
|---|---|
| 1. | Initiate a Class 1 operation invocation from SP A to SP B.<br>Generate an unsuccessful response from SP B to SP A with a valid Invoke ID but a wrong type parameter for this operation. |
| 2. | CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 3. | CHECK B: WAS THE RETURN ERROR COMPONENT PASSED TO TC-USER BY SP A? |
| 4. | CHECK C: WAS THE REJECT COMPONENT SENT BY SP A? |
| 5. | CHECK D: WAS THE INVOCATION 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 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Valid functions; Segmentation for Return Result | | |
| SUBTITLE: Class 1 single operation invocation; IUT as sender: receive segmented components | | |
| PURPOSE: To verify that a single Class 1 operation can be completed by receiving segmented Return Result components | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>2) Arrange the data at SP B 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ——————————→ | |
| | ←—————————— | **RETURN RESULT NOT-LAST (i)** |
| *TC-RESULT-NL ind.*<br><============ | | |
| | ←—————————— | **RETURN RESULT-LAST (i)** |
| *TC-RESULT-L ind.*<br><============ | | |

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 RETURN RESULT NOT-LAST 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

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)

NOTE – Omitted when no parameter is present.

| | |
|---|---|
| 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 invocation; IUT as receiver: send segmented components | |
| PURPOSE: To verify that a single Class 1 operation can be completed by sending segmented 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE ind.* <============ | $\longleftarrow$ | **INVOKE (i)** |
| *TC-RESULT-NL req.* ============> | | |
| **RETURN RESULT NOT-LAST (i)** | $\longrightarrow$ | |
| *TC-RESULT-L req.* ============> | | |
| **RETURN RESULT-LAST (i)** | $\longrightarrow$ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single 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 THE RETURN RESULT NOT-LAST 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: 2.1.5.1.2 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT NOT-LAST component in TSL message from SP B to SP A<br>Component type tag: 10100111 (RETURN RESULT NOT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>RETURN RESULT LAST component in TSL message from SP B to SP A<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.5.2.1 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774

TITLE: Valid functions; Segmentation for Return Result

SUBTITLE: Class 3 single operation invocation; IUT as sender: Receive segmented components

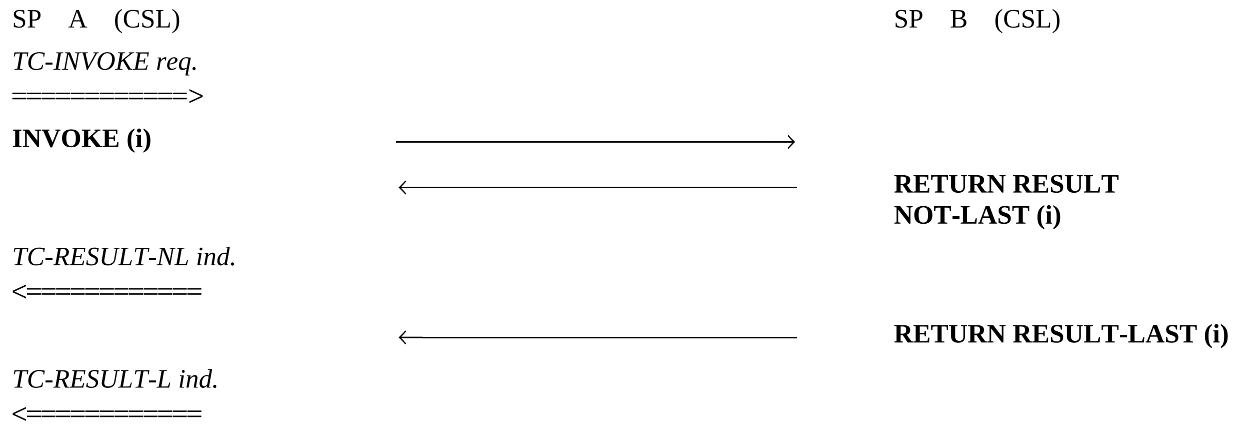
PURPOSE: To verify that a single Class 3 operation can be completed by receiving segmented Return Result components

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 and CPT | TYPE OF SP: SP |

EXPECTED MESSAGE AND COMPONENT FLOW:



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 NOT-LAST 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

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)

NOTE – Omitted when no parameter is present.

| | | |
|---|---|---|
| TEST NUMBER: 2.1.6 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Valid functions | | |
| SUBTITLE: User Cancel | | |
| PURPOSE: To verify that an operation invocation can be canceled by TC-User | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>2) Arrange the data at SP B such that a Return Result-Last 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.*<br>============> | | |
| **INVOKE (i)** | ——————————→ | |
| *TC-U-CANCEL req.*<br>============> | | |
| | ←—————————— | **RETURN-RESULT-LAST (i)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (i)** | ——————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single Class 1 operation invocation from SP A to SP B.<br>Arrange TC-User to cancel the operation immediately after the Invoke component is sent. |
| 2. | CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 3. | CHECK B: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 4. | CHECK C: WAS THE COMPONENT FLOW AS SHOWN ABOVE? |
| 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)

| TEST NUMBER: 2.1.6 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000010 (RETURN RESULT)<br>Problem code length: 00000001<br>Problem code: 00000000 (unrecognized invoke ID)<br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.7.1 | Sheet: 1 of 2 |

REFERENCE: 3.3/Q.773

TITLE: Valid functions; Encoding variations

SUBTITLE: Component length definite short

PURPOSE: To verify that a component portion with a definite short form can be accepted

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 |

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

| | |
|---|---|
| 1. | Initiate a Class 1 or 3 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 THE RETURN RESULT-LAST 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 (definite short form)

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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP A to SP B<br>Component type tag: 10100011 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.7.2 | Sheet: 1 of 2 |

REFERENCE: 3.3/Q.773

TITLE: Valid functions; Encoding variations

SUBTITLE: Component length definite long

PURPOSE: To verify that a component portion with a definite long form can be accepted

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 |

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

| | |
|---|---|
| 1. | Initiate a Class 1 or 3 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 THE RETURN RESULT-LAST 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 (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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP A to SP B<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00000010 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.1.7.3 | Sheet: 1 of 2 |

REFERENCE: 3.3/Q.773

TITLE: Valid functions; Encoding variations

SUBTITLE: Component length indefinite

PURPOSE: To verify that a component portion with a indefinite form can be accepted

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 |

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←———————————— | **INVOKE (i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-RESULT-L req.*<br>============> | | |
| **RETURN RESULT-LAST (i)** | ————————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a Class 1 or 3 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 THE RETURN RESULT-LAST 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 (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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>EOC Tag: 00000000<br>EOC Length: 00000000<br><br>RETURN RESULT-LAST component in TSL message from SP A to SP B<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (see Note)<br><br>parameters (provided by the TC-User)<br><br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| 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 = –127 (FFh) | |
| PURPOSE: To verify that the IUT (SP A) is able to deal with correct encoding of component ID (upper value) | |
| PRE-TEST CONDITIONS: Arrange the TC-User 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 COMPONENT FLOW:

| | | |
|---|---|---|
| SP A (CSL) | | SP B (CSL) |
| | ←―――――――――――――――― | **INVOKE (i)** |
| *TC-INVOKE ind.*<br><============ | | |
| *TC-RESULT-L req.*<br>============> | | |
| **RETURN RESULT-LAST (i)** | ――――――――――――――――→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single operation invocation from SP B to SP A with Invoke ID set to 11111111. |
| 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: 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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP A to SP B<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: 11111111 (FFh)<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.7.4.1.2 | | Sheet: 1 of 2 |
| REFERENCE: 6.2/Q.773 | | |
| TITLE: Valid functions; Encoding variations | | |
| SUBTITLE: Value variations; Invoke ID; Invoke ID = 0 (00h) | | |
| PURPOSE: To verify that the IUT (SP A) is able to deal with correct encoding of component ID (lower value) | | |
| PRE-TEST CONDITIONS: Arrange the TC-User 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 COMPONENT FLOW:

SP A (CSL) SP B (CSL)

$\longleftarrow$ **INVOKE (i)**

*TC-INVOKE ind.*
<============

*TC-RESULT-L req.*
============>

**RETURN RESULT-LAST (i)** $\longrightarrow$

TEST DESCRIPTION

| | |
|---|---|
| 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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP A to SP B<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: 0<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| 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 operation code | | |
| PURPOSE: To verify that a global operation code is correctly decoded by TCAP | | |
| PRE-TEST CONDITIONS: Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains an Invoke component with a global operation code. The global value does not correspond to a supported operation | | |
| 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-U-REJECT req.
============>
REJECT (i)           ――――――――――――――――――――――――――――→
```

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP B to SP A with a non-supported global operation code. |
| 2. | CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION 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 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)

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<br><br>Component type tag: 10100001 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000001 (INVOKE problem type)<br>Problem code length: 00000001<br>Problem code: 00000001 (unrecognized operation) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.8.1 | | Sheet: 1 of 2 |
| REFERENCE: Q.774 | | |
| TITLE: Valid functions; Multiple components grouping | | |
| SUBTITLE: Multiple operations invocation; receiving success | | |
| PURPOSE: To verify that multiple operations can be successfully invoked and the successful completions of the operations can be received | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains multiple components<br>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 |

EXPECTED MESSAGE AND COMPONENT FLOW:

SP A (CSL) SP B (CSL)

*TC-INVOKE req. (#1)*
============>

•

•

•

*TC-INVOKE req. (#n)*
============>

**INVOKE #1, ..., #n**[a)]

$\longrightarrow$

**RETURN RESULT-LAST #1, ..., #n**[a)]

$\longleftarrow$

*TC-RESULT-L ind. (#1)*
<============

•

•

•

*TC-RESULT-L ind. (#n)*
<============

a) The sequence of the components is provided by the TC-User.

NOTE – Number of components is subject to the TC-User.

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate multiple operations within a TSL message from SP A to SP B. |
| 2. | CHECK A: WERE ALL THE INVOKE COMPONENTS WITHIN A TSL MESSAGE SENT BY SP A WITH CORRECT INFORMATION? |
| 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)

NOTE – Omitted when no parameter is present.

| | | |
|---|---|---|
| TEST NUMBER: 2.1.8.2 | | Sheet: 1 of 2 |
| REFERENCE: Q.774 | | |
| TITLE: Valid functions; Multiple components grouping | | |
| SUBTITLE: Multiple operations invocation; reporting success | | |
| PURPOSE: To verify that multiple operations can be successfully invoked and the successful completions of the operations can be sent | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP B contains multiple components<br>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 |

EXPECTED MESSAGE AND COMPONENT FLOW:

SP A (CSL) SP B (CSL)

**INVOKE #1, ..., #n**[a)]
$\longleftarrow$

*TC-INVOKE ind. (#1)*
<============

•

•

•

*TC-INVOKE ind. (#n)*
<============

*TC-RESULT-L req. (#1)*
============>

•

•

•

*TC-RESULT-L req. (#n)*
============>

**RETURN RESULT-LAST #n, ..., #1**[a)]
$\longrightarrow$

a) The sequence of the components is provided by the TC-User.

NOTE – Number of components is subject to the TC-User.

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate multiple operations within a TSL message from SP B to SP A. |
| 2. | 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. | CHECK B: WERE ALL THE RETURN RESULT-LAST COMPONENTS WITHIN A TSL MESSAGE SENT BY SP A WITH CORRECT INFORMATION? |
| 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<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>INVOKE component in TSL message from SP B to SP A<br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: 1, or, ..., n corresponding to the INVOKE #1, ..., #n<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x1, ..., xn representing valid operation codes<br>parameters (provided by the TC-User)<br>RETURN RESULT-LAST component in TSL message from SP A to SP B<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: 1, or, ..., n<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x1, or, ..., xn (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| 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)

――――――――――――――――→

NOTE 1 – The sequence of the Invoke components is important.

NOTE 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<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: 1<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x is a valid operation code<br><br>parameters (provided by the TC-User)<br><br>INVOKE #2 component in TSL message from SP B to SP A<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: 2<br><br>parameters (provided by the TC-User)<br><br>INVOKE #3 component in TSL message from SP B to SP A<br><br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: 3<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x is a valid operation code<br><br>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<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: 1<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x<br><br>parameters (provided by the TC-User)<br>REJECT #2 component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: 2<br><br>Problem code tag: 10000000 (General Problem)<br>Problem code length: 00000001<br>Problem code: 00000010 (badly structured component)<br><br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.1.1 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Accept application context proposal; Send AARQ in Begin message | | |
| PURPOSE: To verify that an IUT can generate and send the dialogue control APDU AARQ within the dialogue portion in a Begin 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 (CSL) SP B (CSL)

*TR-BEGIN req.*
===========>

**BEGIN (AARQ)** $\longrightarrow$

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion. |
| 2. | CHECK A: DOES THE DIALOGUE PORTION IN THE BEGIN MESSAGE CONTAIN THE APDU AARQ?<br>Also arrange an END message to be sent by IUT or by the Tester. When the last message has been sent by Tester, the END message sent by IUT can be used to check that the last message has been correctly received. |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Begin message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
direct reference: H'00118605010101 (structured dialogue abstract syntax)

single ASN.1 type tag: 10100000
single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
direct reference: any object identifier

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.1.2 | | Sheet: 1 of 3 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Accept application context proposal; accept AARQ and continue dialogue | | |
| PURPOSE: To verify that an IUT can receive a Begin message with APDU 'AARQ' and then can generate and send the dialogue control APDU 'AARE' within the dialogue portion of a Continue 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 (CSL) | | SP B (CSL) |
|---|---|---|
| | ⟵ | **BEGIN (AARQ)** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE (AARE)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue request. |
| 2. | CHECK A: DOES THE DIALOGUE PORTION IN THE CONTINUE MESSAGE CONTAIN THE APDU AARE AND IS THE APPLICATION CONTEXT THE SAME AS IN THE RECEIVED AARQ?<br>Also arrange an END message to be sent by IUT or by the Tester. When the last message has been sent by Tester, the END message sent by IUT can be used to check that the last message has been correctly received. |

| TEST NUMBER: 2.1.9.1.2 | Sheet: 2 of 3 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Begin message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101<br>Single ASN.1 type tag: 10100000 (see Note)<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Request tag: 01100000<br>Dialogue Request length: correct number of octets<br>Protocol Version tag: 10000000<br>Protocol Version length: 00000010<br>Protocol Version value: 00000111 10000000<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: any object identifier<br>NOTE – Instead of the encoding option Single ASN.1 type, this test can also be done with the 2 other options – octet aligned and arbitrary. | |

| TEST NUMBER: 2.1.9.1.2 | Sheet: 3 of 3 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Continue message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H’00118605010101<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Response tag: 01100001<br>Dialogue Response length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: same octets as in AARQ<br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000000 (Accepted)<br>Result source diagnostic tag: 10100011<br>Result source diag. length: 00000101<br>Dialogue Service User tag: 10100001<br>Dialogue Service User length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service user value: 00000000 (NULL) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.1.3 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Accept application context proposal; accept AARQ and end dialogue | | |
| PURPOSE: To verify that an IUT can receive a Begin message with APDU AARQ and then can generate and send the dialogue control APDU 'AARE' within the dialogue portion of an End 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 (CSL) | | SP B (CSL) |
|---|---|---|
| | ←———————————— | **BEGIN (AARQ)** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-END req.*<br>============> | | |
| **END (AARE)** | ————————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue request |
| 2. | CHECK A: DOES THE DIALOGUE PORTION IN THE END MESSAGE CONTAIN THE APDU AARE AND IS THE APPLICATION CONTEXT THE SAME AS IN THE RECEIVED AARQ? |

| TEST NUMBER: 2.1.9.1.3 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in End message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H’00118605010101 (structured dialogue abstract syntax)<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Response tag: 01100001<br>Dialogue Response length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: same octets as in AARQ<br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000000 (Accepted)<br>Result source diagnostic tag: 10100011<br>Result source diag. length: 00000101<br>Dialogue Service User tag: 10100001<br>Dialogue Service User length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service user value: 00000000 (NULL) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.2.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Propose alternative application context; Send AARE with alternative | | |
| PURPOSE: To verify that an IUT can generate and send dialogue control APDU ’AARE’ with an alternative application context within the dialogue portion of a Continue 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 (CSL) | | SP B (CSL) |
| | ←―――――――――――――― | **BEGIN (AARQ)** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE (AARE)** | ――――――――――――――→ | |

| | |
|---|---|
| | TEST DESCRIPTION |
| 1. | Arrange for SP B to send a Begin message containing a dialogue request. |
| 2. | Arrange for SP A to propose an alternative application context. |
| 3. | CHECK A: DOES THE DIALOGUE PORTION IN THE CONTINUE MESSAGE CONTAIN THE APDU AARE AND IS THE APPLICATION CONTEXT DIFFERENT THAN THE ONE IN THE RECEIVED AARQ?<br>Also arrange an END message to be sent by IUT or by the Tester. When the last message has been sent by Tester, the END message sent by IUT can be used to check that the last message has been correctly received. |

| TEST NUMBER: 2.1.9.2.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Continue message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101 (structured dialogue abstract syntax)<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Response tag: 01100001<br>Dialogue Response length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: correct number of octets different than those in AARQ<br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000000 (Accepted)<br>Result source diagnostic tag: 10100011<br>Result source diag. Length: 00000101<br>Dialogue Service User tag: 10100001<br>Dialogue Service User length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service user value: 00000000 (NULL) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.2.2 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Propose alternative application context; receive AARE with alternative | | |
| PURPOSE: To verify that an IUT can accept the dialogue control APDU ’AARE’ with an alternative application context within the dialogue portion of a Continue 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *TR-BEGIN req.*<br>============> | | |
| **BEGIN (AARQ)** | ——————————→ | |
| | ←—————————— | **CONTINUE (AARE)** |
| *TR-CONTINUE ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion |
| 2. | Arrange for SP B to confirm the dialogue proposing an alternative application context |
| 3. | CHECK A: DOES THE IUT ACCEPT THE APDU ’AARE’ WITH THE ALTERNATIVE APPLICATION CONTEXT?<br>Also arrange an END message to be sent by IUT or by the Tester. When the last message has been sent by Tester, the END message sent by IUT can be used to check that the last message has been correctly received. |

| TEST NUMBER: 2.1.9.2.2 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Continue message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101 (structured dialogue abstract syntax)<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Response tag: 01100001<br>Dialogue Response length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: correct number of octets different than those in AARQ<br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000000 (Accepted)<br>Result source diagnostic tag: 10100011<br>Result source diag. Length: 00000101<br>Dialogue Service User tag: 10100001<br>Dialogue Service User length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service user value: 00000000 (NULL) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.3 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Dialogue refused | | |
| PURPOSE: To verify that an IUT can generate and send dialogue control APDU 'AARE' within the dialogue portion in an Abort message to indicate that the application context is not supported | | |
| 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 (CSL) | | SP B (CSL) |
| | ←―――――――――――――― | **BEGIN (AARQ)** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR- U - ABORT req.*<br>============> | | |
| **ABORT (AARE)** | ――――――――――――――→ | |

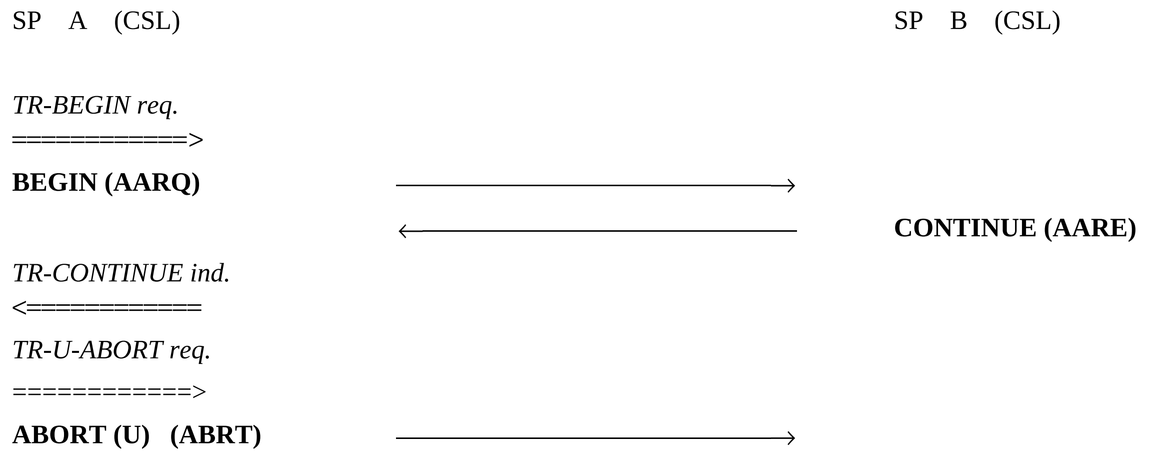
TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue request. |
| 2. | Arrange for SP A to refuse the dialogue because of application context not supported. |
| 3. | CHECK A: DOES SP A TRANSMIT THE EXPECTED ABORT MESSAGE WITH APDU 'AARE'? |

| TEST NUMBER: 2.1.9.3 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Abort message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101 (structured dialogue abstract syntax )<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Response tag: 01100001<br>Dialogue Response length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: same octets as in dialogue request<br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000001 (Reject -Permanent)<br>Result source diagnostic tag: 10100011<br>Result source length: 00000101<br>Dialogue Service User tag: 10100001<br>Dialogue Service User length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service user value: 00000010 (Application Context Not Supported) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.4 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Dialogue abort | | |
| PURPOSE: To verify that an IUT can generate and send the dialogue control APDU 'ABRT' within the dialogue portion in an Abort message after the dialogue has been established | | |
| 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:



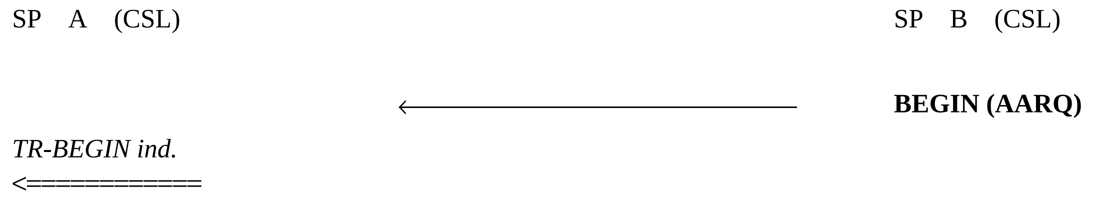
TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion |
| 2. | Arrange for SP B to confirm the dialogue |
| 3. | Arrange for SP A to Abort the dialogue for some reason. |
| 4. | CHECK A: DOES THE DIALOGUE PORTION IN THE ABORT MESSAGE CONTAIN APDU 'ABRT'? |

| TEST NUMBER: 2.1.9.4 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Abort message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101 (structured dialogue abstract syntax )<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Abort tag: 01100100<br>Dialogue Abort length: 00000011<br>Abort Source tag: 10000000<br>Abort source length: 00000001<br>Abort source: 00000000 (dialogue service user) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.5.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Transport of User information; accept user information in Begin message | | |
| PURPOSE: To verify that an IUT can receive a Begin message with APDU AARQ including an user information element | | |
| 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:



TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue request containing user information. |
| 2. | CHECK A: DOES THE IUT ACCEPT USER INFORMATION IN THE RECEIVED DIALOGUE REQUEST?<br>Also arrange an END message to be sent by IUT. The END message sent by IUT can be used to check that the last message has been correctly received. |

| TEST NUMBER: 2.1.9.5.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Begin message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101 (structured dialogue abstract syntax )<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Request tag: 01100000<br>Dialogue Request length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: any object identifier<br>User information in dialogue PDU<br>User Information tag: 10111110<br>User information length: correct number of octets<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: any object identifier<br>Single ASN.1 type tag: 10100000 (see Note)<br>Single ASN.1 type length: correct number of octets<br>Some bytes of user data in the user defined abstract syntax<br>NOTE – This test can also be done with the 2 other encoding options – octet aligned and arbitrary. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.5.2 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Transport of User information; accept user information in first Continue message | | |
| PURPOSE: To verify that an IUT can accept a Continue message with APDU ’AARE’ including a user information element | | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN (AARQ)** | ⟶ | |
| | ⟵ | **CONTINUE (AARE)** |
| *TR-CONTINUE ind.*<br><=========== | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion. |
| 2. | Arrange for SP B to confirm the dialogue including user information in the dialogue portion. |
| 3. | CHECK A: DOES THE IUT ACCEPT USER INFORMATION IN THE RECEIVED DIALOGUE RESPONSE?<br>Also arrange an END message to be sent by IUT or by the tester. When the last message has been sent by the tester, the END message sent by IUT can be used to check that the last message has been correctly received. |

| TEST NUMBER: 2.1.9.5.2 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |

Dialogue portion in Continue message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H'00118605010101 (structured dialogue abstract syntax )

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Response tag: 01100001
Dialogue Response length: correct number of octets

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

Result tag: 10100010
Result length: 00000011
INTEGER type tag: 00000010
INTEGER length: 00000001
Result value: 00000000 (Accepted)

Result source diagnostic tag: 10100011
Result source diag. Length: 00000101
Dialogue Service User tag: 10100001
Dialogue Service User length: 00000011
INTEGER type tag: 00000010
INTEGER length: 00000001
Dialogue service user value: 00000000 (NULL)

User information in dialogue PDU

User Information tag: 10111110
User information length: correct number of octets

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Some bytes of user data in the user defined abstract syntax

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.5.3 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Transport of User information; accept user information in subsequent Continue message | | |
| PURPOSE: To verify that SP A can accept a user information element in a subsequent Continue message | | |
| PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state and test cases 2.1.9.5.2 has to be executed successfully | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←―――――――――――――――― | **BEGIN (AARQ)** |
| *TR-BEGIN ind.*<br><============ | | |
| *TR-CONTINUE req.*<br>============> | | |
| **CONTINUE (AARE)** | ――――――――――――――――→ | |
| | ←―――――――――――――――― | **CONTINUE** |
| *TR-CONTINUE ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message with dialogue request to SP A. |
| 2. | Arrange for SP A to confirm the dialogue. |
| 3. | Arrange for SP B to send a Continue message including an user information element as dialogue portion. |
| 4. | CHECK A: VERIFY THAT THE IUT AT SP A ACCEPTED THE USER INFORMATION ELEMENT |
| | Also arrange an END message to be sent by IUT or by the Tester. When the last message has been sent by Tester, the END message sent by IUT can be used to check that the last message has been correctly received. |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Continue message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion (user information element)

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any abstract syntax

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Some bytes of user data in the user defined abstract syntax

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.5.4 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Transport of User information, accept several user information elements in Continue message | | |
| PURPOSE: To verify that an IUT can accept a Continue message with APDU ’AARE’ including several user information elements | | |
| PRE-TEST CONDITIONS: SP A ( TSL) and SP B (TSL) are to be in the idle state and test cases 2.1.9.5.2 has to be executed successfully | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT and CPT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN (AARQ)** | ⟶ | |
| | ⟵ | **CONTINUE (AARE)** |
| *TR-CONTINUE ind.*<br><=========== | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion. |
| 2. | Arrange for SP B to confirm the dialogue including user information elements in the dialogue portion. |
| 3. | CHECK A: DOES THE IUT ACCEPT USER INFORMATION IN THE RECEIVED DIALOGUE RESPONSE?<br>Also arrange an END message to be sent by IUT or by the Tester. When the last message has been sent by Tester, the END message sent by IUT can be used to check that the last message has been correctly received. |

| TEST NUMBER: 2.1.9.5.4 | Sheet: 2 of 2 |
|---|---|

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Continue message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H'00118605010101

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Response tag: 01100001
Dialogue Response length: correct number of octets

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

Result tag: 10100010
Result length: 00000011
INTEGER type tag: 00000010
INTEGER length: 00000001
Result value: 00000000 (Accepted)

Result source diagnostic tag: 10100011
Result source diag. Length: 00000101
Dialogue Service User tag: 10100001
Dialogue Service User length: 00000011
INTEGER type tag: 00000010
INTEGER length: 00000001
Dialogue service user value: 00000000 (NULL)

User information in dialogue PDU

User Information tag: 10111110
User information length: correct number of octets

External type tag: 00101000 (external number 1)
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any user defined abstract syntax

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Some bytes representing element 1

External type tag: 00101000 (external number 2)
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any user defined abstract syntax

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Some bytes of user data in user information element number 2

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.6 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Unstructured dialogue | | |
| PURPOSE: To verify that the IUT can accept the dialogue control APDU ’AUDT’ in an Unidirectional 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 (CSL) SP B (CSL)

$\longleftarrow$ **UNIDIRECTIONAL (AUDT)**

*TR-UNI ind.*

<============

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange SP B to send an Unidirectional message containing a dialogue portion to SP A. |
| 2. | CHECK A: WAS THE UNIDIRECTIONAL MESSAGE WITH APDU ’AUDT’ CORRECTLY RECEIVED AT SP A? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

Dialogue portion in Unidirectional message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H’00118605010201

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.7.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.3/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Dialogue control APDU Version Structured dialogue; Version not 1 | | |
| PURPOSE: To verify that an IUT can abort the dialogue if the first bit of the protocol version field in the dialogue request, is not set to 1. The dialogue has to be aborted with APDU 'AARE' (reject permanent, no common dialogue) | | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| | ←―――――――――――― | **BEGIN (AARQ)** |
| *TR-U-ABORT req.*<br>============><br>**ABORT (AARE)** | ――――――――――――→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue request indicating that Version 1 is not supported |
| 2. | CHECK A: DOES SP A TRANSMIT THE EXPECTED ABORT MESSAGE WITH APDU 'AARE' INDICATING NO COMMON DIALOGUE PORTION? |

| TEST NUMBER: 2.1.9.7.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Abort message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Response tag: 01100001<br>Dialogue Response length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: same octets as in dialogue request<br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000001 (Reject-Permanent)<br>Result source diagnostic tag: 10100011<br>Result source length: 00000101<br>Dialogue Service Provider tag: 10100010<br>Dialogue Service Provider length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service provider value: 00000010 (No common dialogue portion) | |

| | |
|---|---|
| TEST NUMBER: 2.1.9.7.2 | Sheet: 1 of 1 |

REFERENCE: 3.2.3/Q.774

TITLE: Valid functions; Dialogue Portion

SUBTITLE: Dialogue control APDU Version Structured dialogue; Version 1

PURPOSE: To verify that an IUT can accept a dialogue request offering several versions including version 1. The IUT response must be of version 1

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 (CSL) | | SP B (CSL) |
|---|---|---|
| | ←―――――――――――――――― | **BEGIN (AARQ-Vx)** |
| *TR-BEGIN ind.* <============ | | |
| *TR-CONTINUE req.* ============> | | |
| **CONTINUE (AARE-V1)** | ――――――――――――――――→ | |

TEST DESCRIPTION

1. Arrange for SP B to send a Begin message containing a dialogue request offering several versions including Version 1.
2. CHECK A: DOES THE DIALOGUE PORTION IN THE CONTINUE MESSAGE CONTAIN THE APDU AARE AND IS IT OF VERSION 1?

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Begin message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H'00118605010101

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Protocol Version tag: 10000000
Protocol Version length: 00000010
Protocol Version: 00000110 11000000 (versions 1 and 2 supported)

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

| | |
|---|---|
| TEST NUMBER: 2.1.9.7.3 | Sheet: 1 of 1 |
| REFERENCE: 3.2.1.2/Q.774 | |
| TITLE: Valid functions; Dialogue Portion | |
| SUBTITLE: Dialogue control APDU Version Unstructured dialogue; Version not 1 | |
| PURPOSE: To verify that an IUT can discard a UNIDIRECTIONAL msg if the first bit of the protocol version field in the dialogue request AUDT, is not set to 1 | |
| PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state and test case 2.1.9.6 has to be executed successfully | |

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

EXPECTED MESSAGE SEQUENCE:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *msg discarded* | $\longleftarrow$ | **UNIDIRECTIONAL (AUDT)** |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange SP B to send an Unidirectional msg containing APDU AUDT indicating that Version 1 is not supported to SP A. |
| 2. | CHECK A: WAS THE UNIDIRECTIONAL MESSAGE WITH APDU ’AUDT’ DISCARDED AT SP A? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

Dialogue portion in Unidirectional message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H’00118605010201

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Protocol Version tag: 10000000
Protocol Version length: 00000010
Protocol Version: 00000110 01000000 (only version 2 supported)

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

| | | |
|---|---|---|
| TEST NUMBER: 2.1.9.7.4 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Valid functions; Dialogue Portion | | |
| SUBTITLE: Dialogue control APDU Version Unstructured dialogue; Version 1 | | |
| PURPOSE: To verify that an IUT can accept a UNIDIRECTIONAL msg if the first bit of the protocol version is set to 1 and also other bits are set to 1 | | |
| PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state and test case 2.1.9.6 has to be executed successfully | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←―――――――――――――― | **UNIDIRECTIONAL (AUDT)** |
| *TR-UNI ind.*<br><============ | | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange SP B to send an Unidirectional msg containing APDU AUDT offering several versions including Version 1, to SP A. |
| 2. | CHECK A: WAS THE UNIDIRECTIONAL MESSAGE WITH APDU 'AUDT' ACCEPTED AT SP A? |

| TEST NUMBER: 2.1.9.7.4 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES | |
| Dialogue portion in Unidirectional message<br><br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br><br>External data type in dialogue portion<br><br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010201<br><br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br><br>Dialogue PDU<br><br>Dialogue Request tag: 01100000<br>Dialogue Request length: correct number of octets<br><br>Protocol Version tag: 10000000<br>Protocol Version length: 00000010<br>Protocol Version: 00000110 11000000 (version 1 and version 2 supported)<br><br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: any object identifier | |

| | | |
|---|---|---|
| TEST NUMBER: 2.2.1.1 | | Sheet: 1 of 2 |
| REFERENCE: 6.2/Q.773 | | |
| TITLE: Syntactically invalid behaviour; Invalid values for information elements | | |
| SUBTITLE: Length of Invoke ID >1 in Invoke component | | |
| PURPOSE: To verify that a rejection of a requested operation can be performed due to incorrect encoding of component ID (value out of range) | | |
| 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:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←―――――――――――――――― | **INVOKE (i)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (NULL)** | ――――――――――――――――→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP B to SP A with Invoke ID equal to 2 octets (illegal value). |
| 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 octets)
Invoke ID: 129

| TEST NUMBER: 2.2.1.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>NULL tag: 00000101<br>NULL length: 00000000<br>Problem code tag: 10000000 (General problem type)<br>Problem code length: 00000001<br>Problem code: 00000001 (mistyped component) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.2.1.2 | | Sheet: 1 of 1 |
| REFERENCE: 6.2/Q.773 | | |
| TITLE: Syntactically invalid behaviour; Invalid values for information elements | | |
| SUBTITLE: Length of Invoke ID = 0 in Invoke component | | |
| PURPOSE: To verify that a rejection of a requested operation can be performed due to incorrect encoding of component ID (length equals 0) | | |
| 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:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←———————— | **INVOKE** |
| *TC-L-REJECT ind.* <============ | | |
| **REJECT (NULL)** | ————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP B to SP A with Invoke ID equal to 0 octets (illegal value). |
| 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: 00000000 (zero octet)

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

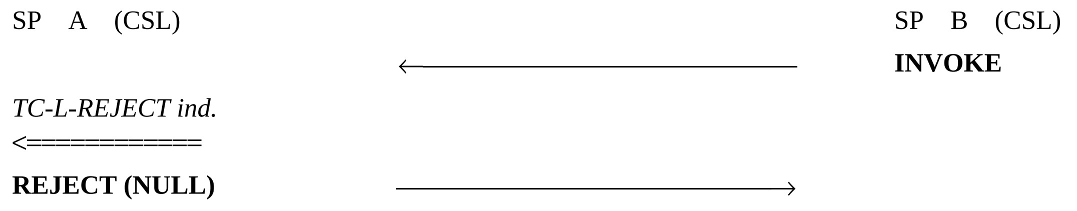
Component type tag: 10100100 (REJECT)
Component length: correct number of octets

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 1 |
| REFERENCE: 6.2/Q.773 | | |
| TITLE: Syntactically invalid behaviour; 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 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:



TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single operation invocation from SP B to SP A with Invoke ID missing. |
| 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

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)

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 1 |

REFERENCE: 3.2.2.2/Q.774

TITLE: Syntactically invalid behaviour; Invalid structure

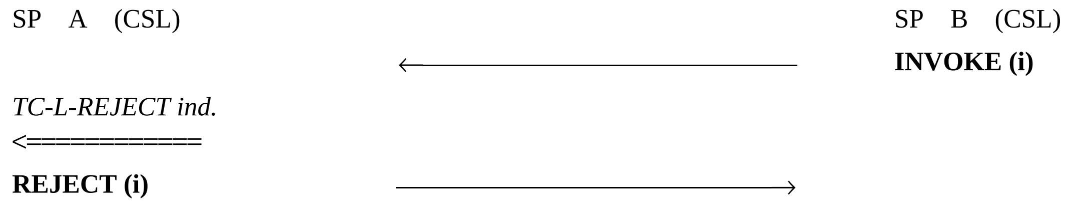
SUBTITLE: Invoke component; Operation code missing

PURPOSE: To verify that a rejection of a requested operation can be performed due to operation code missing

PRE-TEST CONDITIONS: Arrange the stimulus such that an appropriate TSL message generated at SP B contains an Invoke component with a syntax error as described below

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

EXPECTED MESSAGE AND COMPONENT FLOW:



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 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 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)

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 behaviour; Invalid structure

SUBTITLE: Return Result component; Invoke ID missing

PURPOSE: To verify that a rejection can be successfully initiated due to the absence of the Invoke ID in the Return Result-Last component

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 without an Invoke ID is generated

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

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **RETURN RESULT-LAST** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (NULL)**<br>time expiry for invocation (i) | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a Class 1 or 3 operation invocation from SP A to SP B.<br>Generate a response from SP B to SP A without an 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. | 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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: y (y is different from x) (see Note)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>NULL tag: 00000101<br>NULL length: 00000000<br><br>Problem code tag: 10000000 (General problem type)<br>Problem code length: 00000001<br>Problem code: 00000001 (wrong type component)<br><br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.2.2.2.2 | Sheet: 1 of 2 |

REFERENCE: 3.2.2/Q.774

TITLE: Syntactically invalid behaviour; Invalid structure

SUBTITLE: Return Result component; Operation code missing while parameters included

PURPOSE: To verify that a rejection can be successfully initiated due to the operation code being missing in the Return Result-Last component

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 3
2) Arrange the data at SP B such that a Return Result-Last without an operation code is generated

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

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **RETURN RESULT-LAST (i)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (i)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP A to SP B.<br>Generate a response from SP B to SP A with a valid Invoke ID but a different operation code. |
| 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. | 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: 2.2.2.2.2 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000<br>Sequence length: correct number of octets<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000000 (General problem type)<br>Problem code length: 00000001<br>Problem code: 00000001 (wrong type component) | |

| TEST NUMBER: 2.2.2.2.3 | Sheet: 1 of 2 |
|---|---|

REFERENCE: 6.4/Q.773; 3.2.2.2/Q.774

TITLE: Syntactically invalid behaviour; Invalid structure

SUBTITLE: Return Result component; Sequence tag missing while parameters included

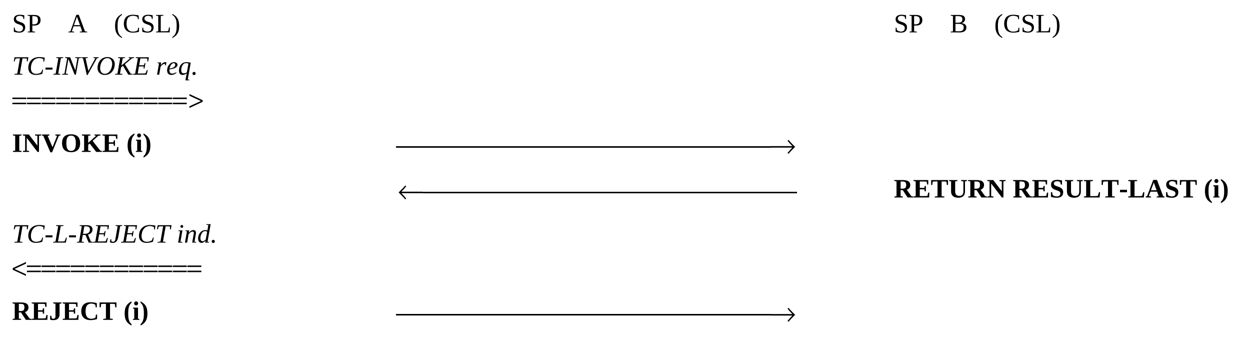
PURPOSE: To verify that a rejection can be successfully initiated due to Sequence tag missing while parameters included

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 COMPONENT FLOW:



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 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: 2.2.2.2.3 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000000 (General problem type)<br>Problem code length: 00000001<br>Problem code: 00000001 (wrong type component) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.2.2.3.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Return Error; Invoke ID missing | | |
| PURPOSE: To verify that a rejection can be successfully initiated due to the absence of the Invoke ID in the Return Error component | | |

PRE-TEST CONDITIONS:

1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component of the Class 1
2) Arrange the data at SP B such that a Return Error without an Invoke ID is generated

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

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ──────────────────→ | |
| | ←────────────────── | **RETURN ERROR** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (NULL)**<br>time expiry for invocation (i) | ──────────────────→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a Class 1 operation invocation from SP A to SP B.<br>Generate an unsuccessful response from SP B to SP A without an 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. | 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: 2.2.2.3.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN ERROR component in TSL message from SP B to SP A<br>Component type tag: 10100010 (RETURN ERROR)<br>Component length: correct number of octets<br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y (y is an error code which the invoked operation may report)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>NULL tag: 00000101<br>NULL length: 00000000<br>Problem code tag: 10000000 (General problem type)<br>Problem code length: 00000001<br>Problem code: 00000001 (mistyped component) | |

| | |
|---|---|
| TEST NUMBER: 2.2.2.3.2 | Sheet: 1 of 2 |

REFERENCE: 3.2.2/Q.774

TITLE: Syntactically invalid behaviour; Invalid structure

SUBTITLE: Return Error; Error code missing

PURPOSE: To verify that a rejection can be successfully initiated due to the absence of the error code in the Return Error component

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 without an error code is generated

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

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.* <br> ============> | | |
| **INVOKE (i)** | ——————————→ | |
| | ←—————————— | **RETURN ERROR (i)** |
| *TC-L-REJECT ind.* <br> <============ | | |
| **REJECT (i)** | ——————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a Class 1 operation invocation from SP A to SP B.<br>Generate an unsuccessful response from SP B to SP A with a valid Invoke ID but without error code for this operation. |
| 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. | 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: 2.2.2.3.2 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN ERROR component in TSL message from SP B to SP A<br><br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000101<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000000 (General problem type)<br>Problem code length: 00000001<br>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 behaviour; 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.*<br><============ | | |
| **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 1 |
| REFERENCE: 3.2.2.2/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Unknown component type; Invoke ID derivable | | |
| PURPOSE: To verify that a rejection can be initiated due to Unknown component type with derivable Invoke ID | | |
| PRE-TEST CONDITIONS: Arrange the stimulus such that an appropriate TSL message generated at SP B contains an Unknown component with a derivable Invoke ID 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 (i) |
| *TC-L-REJECT ind.*<br><=========== | | |
| **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)

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

or

NULL tag: 00000101
NULL length: 00000000

Problem code tag: 10000000 (General problem type)
Problem code length: 00000001
Problem code: 00000000 (unrecognized component)

| | |
|---|---|
| TEST NUMBER: 2.2.2.5.1 | Sheet: 1 of 2 |

REFERENCE: 3.2.2.1/Q.774

TITLE: Syntactically invalid behaviour; Invalid structure

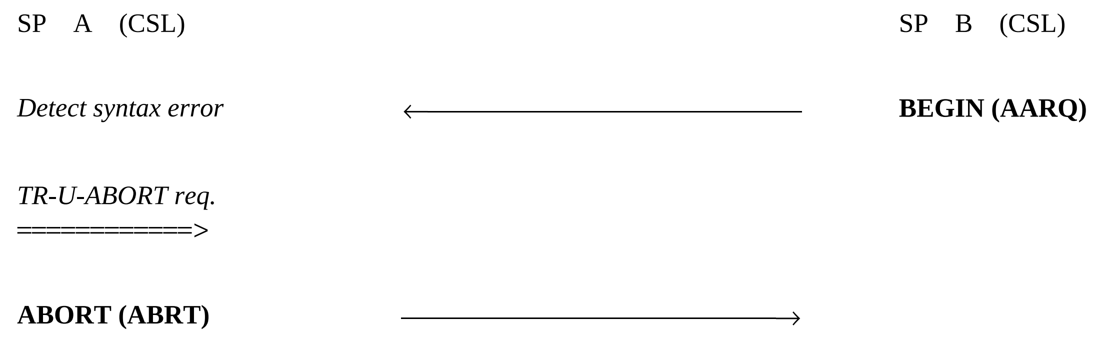
SUBTITLE: Dialogue Portion; Missing Application Context in APDU AARQ

PURPOSE: To verify that the IUT aborts the transaction upon reception of a Begin message containing an APDU 'AARQ' without application context parameter.

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:



TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue request without application context. |
| 2. | CHECK A: DOES SP A TRANSMIT THE EXPECTED ABORT MESSAGE WITH APDU 'ABRT'? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Abort message

Dialogue portion tag: 01101011
Dialogue portion length: 00010010

External data type in dialogue portion

External type tag: 00101000
External length: 00010000

Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H'00118605010101 (structured dialogue abstract syntax)

Single ASN.1 type tag: 10100000
Single ASN.1 type length: 00000101

Dialogue PDU

Dialogue Abort tag: 01100100
Dialogue Abort length: 00000011

Abort Source tag: 10000000
Abort source length: 00000001
Abort source: 00000001 (dialogue service provider)

| TEST NUMBER: 2.2.2.5.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Begin message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H’00118605010101<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Request tag: 01100000<br>Dialogue Request length: correct number of octets<br>Application context tag: Missing | |

| | |
|---|---|
| TEST NUMBER: 2.2.2.5.2 | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.1/Q.774 | |
| TITLE: Syntactically invalid behaviour; Invalid structure | |
| SUBTITLE: Dialogue Portion; Incorrect length | |
| PURPOSE: To verify that the IUT aborts the transaction upon reception of a Continue message containing an APDU 'AARE' with an incorrect AARE length indicator | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN (AARQ)** | ⟶ | |
| *Detect syntax error* | ⟵ | **CONTINUE (AARE)** |
| *TR-P-ABORT ind.*<br><=========== | | |
| *TR-U-ABORT req.*<br>===========> | | |
| **ABORT (ABRT)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion |
| 2. | Arrange for SP B to confirm the dialogue with an incorrect AARE length indicator in the APDU AARE. |
| 3. | CHECK A: DOES THE DIALOGUE PORTION IN THE ABORT MESSAGE CONTAIN APDU 'ABRT'? |
| 4. | CHECK B: VERIFY THAT THE DIALOGUE AT SP A HAS BEEN TERMINATED. |

| TEST NUMBER: 2.2.2.5.2 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Continue message<br><br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br><br>External data type in dialogue portion<br><br>External type tag: 00101000<br>External length: correct number of octets<br><br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101<br><br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br><br>Dialogue PDU<br><br>Dialogue Response tag: 01100001<br>Dialogue Response length: 01111111 ( incorrect)<br><br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: same octets as in dialogue request<br><br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000000 (Accepted)<br><br>Result source diagnostic tag: 10100011<br>Result source length: 00000101<br>Dialogue Service User tag: 10100001<br>Dialogue Service User length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service user value: 00000000 ( Null) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.2.2.5.3 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Dialogue Portion; Missing result-source-diagnostic | | |
| PURPOSE: To verify that the IUT aborts the transaction upon reception of a Continue message containing an APDU ’AARE’ with a missing result-source-diagnostic parameter | | |
| 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 (CSL) | | SP B (CSL) |
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN (AARQ)** | ——————————→ | |
| *Detect syntax error* | ←—————————— | **CONTINUE (AARE)** |
| *TR-P-ABORT ind.*<br><=========== | | |
| *TR-U-ABORT req.*<br>===========> | | |
| **ABORT (ABRT)** | ——————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion. |
| 2. | Arrange for SP B to confirm the dialogue with an incorrect AARE, parameter missing. |
| 3. | CHECK A: VERIFY THAT THE DIALOGUE AT SP A HAS BEEN TERMINATED. |
| 4. | CHECK B: DOES THE DIALOGUE PORTION IN THE ABORT MESSAGE CONTAIN APDU ’ABRT’? |

| TEST NUMBER:   2.2.2.5.3 | Sheet:   2 of 2 |
|---|---|

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Continue message

Dialogue portion tag:   01101011
Dialogue portion length:   correct number of octets

External data type in dialogue portion

External type tag:   00101000
External length:   correct number of octets

Object Identifier tag:   00000110
Object Identifier length:   00000111
Direct reference:   H'00118605010101

Single ASN.1 type tag:   10100000
Single ASN.1 type length:   correct number of octets

Dialogue PDU

Dialogue Response tag:   01100001
Dialogue Response length:   correct number of octets

Application context tag:   10100001
Application context length:   correct number of octets
Object Identifier tag:   00000110
Object Identifier length:   correct number of octets
Direct reference:   same octets as in dialogue request

Result tag:   10100010
Result length:   00000011
INTEGER type tag:   00000010
INTEGER length:   00000001
Result value:   00000000 (Accepted)

Result source diagnostic tag:   Missing

| | |
|---|---|
| TEST NUMBER: 2.2.2.5.4 | Sheet: 1 of 1 |
| REFERENCE: 3.2.2.1/Q.774 | |
| TITLE: Syntactically invalid behaviour; Invalid structure | |
| SUBTITLE: Dialogue Portion; Missing application context in APDU AUDT | |
| PURPOSE: To verify that the IUT discards an Unidirectional message containing a dialogue request without AC parameter | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *Detect syntax error* | ←——————————— | **UNIDIRECTIONAL (AUDT)** |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send an Unidirectional message with corrupted dialogue request to SP A. |
| 2. | CHECK A: VERIFY THAT NO MESSAGE IS GENERATED IN RESPONSE TO THE RECEIVED MESSAGE. |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue Portion in Unidirectional message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H’00118605010201

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Protocol Version tag: 10000000
Protocol Version length: 00000010
Protocol Version: 00000110
11000000

Application context tag: missing

| | | |
|---|---|---|
| TEST NUMBER: 2.2.2.5.5 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Dialogue Portion; External type without direct reference | | |
| PURPOSE: To verify that the IUT aborts the transaction on reception of a Begin message containing an external type which does not contain a direct reference | | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *Detect semantic error* | ←———————————— | **BEGIN (AARQ)** |
| *TR-U-ABORT req.*<br>===========> | | |
| **ABORT (ABRT)** | ————————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue portion with an external type without direct reference. |
| 2. | CHECK A: DOES SP A TRANSMIT THE EXPECTED ABORT MESSAGE WITH APDU 'ABRT'? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Begin message.

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: missing
Direct reference: missing

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

ANY number of bytes

| | | |
|---|---|---|
| TEST NUMBER: 2.2.2.5.6 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Dialogue Portion; Indirect reference in External type | | |
| PURPOSE: To verify that the IUT aborts the transaction on reception of a Begin message containing an external type which contains both a direct reference and indirect reference | | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *Detect semantic error* | ←———————————— | **BEGIN (AARQ)** |
| *TR-U-ABORT req.*<br>===========> | | |
| **ABORT (ABRT)** | ————————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue portion with an external type also containing an indirect reference. |
| 2. | CHECK A: DOES SP A TRANSMIT THE EXPECTED ABORT MESSAGE WITH APDU ’ABRT’? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Begin message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H’00118605010101
INTEGER type tag: 00000010
INTEGER length: 00000001
Indirect reference: 00000001

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

| | |
|---|---|
| TEST NUMBER: 2.2.2.5.7 | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.1/Q.774 | |
| TITLE: Syntactically invalid behaviour; Invalid structure | |
| SUBTITLE: Dialogue Portion; User information without direct reference | |
| PURPOSE: To verify that the IUT aborts a dialogue on reception of an User information element without direct reference | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *Detect semantic error* | $\longleftarrow$ | **BEGIN (AARQ)** |
| *TR-U-ABORT req.* ===========> | | |
| **ABORT (ABRT)** | $\longrightarrow$ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue portion including a user information element whose external type does not contain a direct reference. |
| 2. | CHECK A: DOES SP A TRANSMIT THE EXPECTED ABORT MESSAGE WITH APDU 'ABRT'? |

| TEST NUMBER: 2.2.2.5.7 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Begin message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Request tag: 01100000<br>Dialogue Request length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: any object identifier<br>User information in dialogue PDU<br>User Information tag: 10111110<br>User information length: correct number of octets<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: missing<br>Object Identifier length: missing<br>Direct reference: missing<br>Octet-aligned tag: 10000001<br>Octet aligned length: correct number of octets<br>Octet string value: any number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 2.2.2.5.8 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid structure | | |
| SUBTITLE: Dialogue Portion; Indirect reference in User information | | |
| PURPOSE: To verify that the IUT aborts a dialogue on reception of an User information element which contains both a direct reference and an indirect reference | | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *Detect semantic error* | ←———————————— | **BEGIN (AARQ)** |
| *TR-U-ABORT req.* ============> | | |
| **ABORT (ABRT)** | ————————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue portion with an external type also containing an indirect reference. |
| 2 | CHECK A: DOES SP A TRANSMIT THE EXPECTED ABORT MESSAGE WITH APDU 'ABRT'? |

| TEST NUMBER: 2.2.2.5.8 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Begin message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Request tag: 01100000<br>Dialogue Request length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: any object identifier<br>User information in dialogue PDU<br>User Information tag: 10111110<br>User information length: correct number of octets<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: any object identifier<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Indirect reference: 00000001<br>Octet-aligned tag: 10000001<br>Octet aligned length: correct number of octets<br>Octet string value: any number of octets | |

| | | |
|---|---|---|
| TEST NUMBER: 2.2.3.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.2/Q.774; 3.2.3/Q.773 | | |
| TITLE: Syntactically invalid behaviour; 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 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:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ⟵ | **INVOKE (i)** |
| *TC-L-REJECT ind.* <============ | | |
| **REJECT (i or NULL)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP B to SP A with an invalid tag. |
| 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: 00100010<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>or<br>NULL tag: 00000101<br>NULL length: 00000000<br>Problem code tag: 10000000 (General problem type)<br>Problem code length: 00000001<br>Problem code: 00000010 (badly structured component) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.2.3.2 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.2/Q.774 | | |
| TITLE: Syntactically invalid behaviour; Invalid encoding for Invoke component | | |
| SUBTITLE: Wrong component length | | |
| PURPOSE: To verify that a rejection of a requested operation can be initiated due to wrong component length | | |
| PRE-TEST CONDITIONS: Arrange the stimulus such that an appropriate TSL message generated at SP B contains an Invoke component with a syntax error as described below | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE AND COMPONENT FLOW:

SP A (CSL) SP B (CSL)

←―――――――――――――――――― **INVOKE (i)**

*TC-L-REJECT ind.*
<============

**REJECT (i or NULL)** ――――――――――――――――――→

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP B to SP A with an invalid component length value. |
| 2. | CHECK A: WAS THE REJECT COMPONENT WITH CORRECT INFORMATION 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 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)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>or<br><br>NULL tag: 00000101<br>NULL length: 00000000<br><br>Problem code tag: 10000000 (General problem type)<br>Problem code length: 00000001<br>Problem code: 00000010 (badly structured component) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.2.3.3 | | Sheet: 1 of 1 |
| REFERENCE: 3.3/Q.773 | | |
| TITLE: Syntactically invalid behaviour; 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 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ←———————————— | **INVOKE (i)** |
| *TC-L-REJECT ind.* <============ | | |
| **REJECT (i or NULL)** | ————————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a single operation invocation from SP B to SP A. |
| 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 (indefinite form)

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: i

or

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 Behaviour; Inopportune Invoke component

SUBTITLE: Invalid linked ID

PURPOSE: To verify that a rejection of a requested operation can be initiated due to invalid 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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.* ============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **INVOKE (j, k)** |
| *TC-L-REJECT ind.* <============ | | |
| **REJECT (j)** | ⟶ | |
| time expiry for invocation (i) | | |
| *TC-L-CANCEL 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 THE REJECT COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 4. | 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 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

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)<br>Operation code length:   correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code:   x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>INVOKE component in TSL message sent by SP B<br><br>Component type tag:   10100001 (INVOKE)<br>Component length:   correct number of octets<br><br>Invoke ID tag:   00000010<br>Invoke ID length:   00000001 (one octet)<br>Invoke ID:   j (j represents an integer)<br><br>Linked ID tag:   10000000<br>Linked ID length:   00000001 (one octet)<br>Linked ID:   k (k is an integer which is different from i)<br><br>Operation code tag:   00000010 (local) or 00000110 (global)<br>Operation code length:  correct number of octets (e.g. 00000001 if y is one octet long)<br>Operation code:   y (y represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in the TSL message sent by SP A<br><br>Component type tag:   10100100 (REJECT)<br>Component length:   correct number of octets<br><br>Invoke ID tag:   00000010<br>Invoke ID length:   00000001<br>Invoke ID:   j<br><br>Problem code tag:   10000001 (INVOKE)<br>Problem code length:   00000001<br>Problem code:   00000101 (unrecognized linked ID) | |

| | |
|---|---|
| TEST NUMBER: 2.3.2.1 | Sheet: 1 of 2 |

REFERENCE: 3.2.2/Q.774

TITLE: Inopportune Behaviour; Unrecognized Invoke ID

SUBTITLE: 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:

1) 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
2) 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 |

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ————————————→ | |
| | ←———————————— | **RETURN RESULT-LAST (j)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (j)** | ————————————→ | |
| time expiry for invocation (i) | | |
| *TC-L-CANCEL ind.*<br><============ | | |
| | ←———————————— | **RETURN RESULT-LAST (i)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (i)** | ————————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP A to SP B.<br>Generate a response from SP B to 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 component from SP B to SP A. |
| 5. | CHECK C: WAS THE REJECT COMPONENT SENT BY SP A? |
| 6. | CHECK D: WAS THE COMPONENT FLOW AS SHOWN IN ABOVE ? |

| 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 Behaviour; Unrecognized Invoke ID | | |
| SUBTITLE: 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:<br>1) 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<br>2) 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 |

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ——————————→ | |
| | ←—————————— | **RETURN RESULT NOT-LAST (j)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (j)** | ——————————→ | |
| time expiry for invocation (i) | | |
| *TC-L-CANCEL ind.*<br><============ | | |
| | ←—————————— | **RETURN RESULT NOT-LAST (i)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (i)** | ——————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP A to SP B.<br>Generate a response from SP B to 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-Last component from SP B to SP A. |
| 5. | CHECK C: WAS THE REJECT COMPONENT SENT BY SP A? |
| 6. | CHECK D: WAS THE COMPONENT FLOW AS SHOWN IN ABOVE ? |

| TEST NUMBER: 2.3.2.2 | Sheet: 2 of 3 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Component portion in TSL messages<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>INVOKE component in TSL message from SP A to SP B<br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN RESULT NOT-LAST component in TSL message from SP B to SP A<br>Component type tag: 10100111 (RETURN RESULT NOT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: j (j is different from i)<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>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)<br>Problem code length: 00000001<br>Problem code: 00000000 (unrecognized invoke ID)<br>RETURN RESULT NOT-LAST component in TSL message from SP B to SP A<br>Component type tag: 10100111 (RETURN RESULT NOT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Problem code tag: 10000010 (RETURN RESULT)<br>Problem code length: 00000001<br>Problem code: 00000000 (unrecognized invoke ID)<br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.3.2.3 | Sheet: 1 of 3 |

REFERENCE: 3.2.2/Q.774

TITLE: Inopportune Behaviour; Unrecognized Invoke ID

SUBTITLE: Inopportune Return Error 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 Error component

PRE-TEST CONDITIONS:

1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component for unrecognized operation Class 1 or 2
2) Arrange the data at SP B such that a Return Error with an invalid Invoke ID is generated

| | | |
|---|---|---|
| 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)            ---------------------------->
                      <----------------------------   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 from SP A to SP B.<br>Generate an unsuccessful response from SP B to SP A with an invalid 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 Error component from SP B to SP A. |
| 5. | CHECK C: WAS THE REJECT COMPONENT SENT BY SP A? |
| 6. | CHECK D: WAS THE COMPONENT FLOW AS ABOVE ? |

| TEST NUMBER: 2.3.2.3 | Sheet: 2 of 3 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Component portion in TSL messages<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>INVOKE component in TSL message from SP A to SP B<br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>RETURN ERROR component in TSL message from SP B to SP A<br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: j (j is different from i)<br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: j<br>Problem code tag: 10000011 (RETURN ERROR)<br>Problem code length: 00000001<br>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<br><br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000011 (RETURN ERROR)<br>Problem code length: 00000001<br>Problem code: 00000000 (unrecognized invoke ID) | |

| | |
|---|---|
| TEST NUMBER: 2.3.2.4 | Sheet: 1 of 2 |

REFERENCE: 3.2.2/Q.774

TITLE: Inopportune Behaviour; Unrecognized Invoke ID

SUBTITLE: Inopportune Reject component

PURPOSE: To verify that receipt of a Reject component with an Invoke ID not corresponding to any active invocation has no effect on an active invocation

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 |

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.* ============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **REJECT (j)** |
| *TC-R-REJECT ind.*[a)] <============ | | |
| | ⟵ | **RETURN RESULT-LAST (i)** |
| *TC-L-RESULT ind.* <============ | | |

a) The issuing of the TC-R-REJECT ind. is implementation dependent.

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate an operation invocation from SP A to SP B.<br>Generate a reject from SP B to SP A with an invalid Invoke ID. |
| 2. | CHECK A: WAS THE INVOKE COMPONENT WITH CORRECT INFORMATION SENT BY SP A? |
| 3. | Generate a Reject component from SP B to SP A. |
| 4. | CHECK B: WAS THE COMPONENT FLOW AS ABOVE? |
| 5. | CHECK C: WAS THE INVOCATION 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<br>Component portion tag: 01101100<br>Component portion length: correct number of octets<br>INVOKE component in TSL message from SP A to SP B<br>Component type tag: 10100001 (INVOKE)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001 (one octet)<br>Invoke ID: i (i represents an integer)<br>Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br>parameters (provided by the TC-User)<br>REJECT component in TSL message from SP A to SP B<br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: j (j is different from i)<br>Problem code tag: 10000001 (INVOKE)<br>Problem code length: 00000001<br>Problem code: any value<br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br>Operation code tag: 00000010 (local) or 00000101 (Global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br>parameters (provided by the TC-User)<br>NOTE – Omitted when no parameter is present. | |

| | |
|---|---|
| TEST NUMBER: 2.3.3.1 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774

TITLE: Inopportune Behaviour; Unexpected Components

SUBTITLE: Return Result-Last for Class 2

PURPOSE: To verify that a rejection can be sent if a Return Result-Last component 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-Last component can be generated

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

EXPECTED MESSAGE AND COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ——————————→ | |
| | ←—————————— | **RETURN RESULT-LAST (i)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (i)** | ——————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a Class 2 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 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: 2.3.3.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT-LAST component in TSL message from SP B to SP A<br><br>Component type tag: 10100010 (RETURN RESULT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000010 (RETURN RESULT)<br>Problem code length: 00000001<br>Problem code: 00000001 (return result unexpected)<br><br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.3.3.2 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Inopportune Behaviour; Unexpected Components | | |
| SUBTITLE: Return Result-Last for Class 4 | | |
| PURPOSE: To verify that a rejection can be sent if a Return Result-Last component is received for a Class 4 operation | | |
| PRE-TEST CONDITIONS:<br>1) Arrange the TC-User stimulus such that an appropriate TSL message generated at SP A contains an Invoke component<br>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 COMPONENT FLOW:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| *TC-INVOKE req.*<br>============> | | |
| **INVOKE (i)** | ⟶ | |
| | ⟵ | **RETURN RESULT-LAST (i)** |
| *TC-L-REJECT ind.*<br><============ | | |
| **REJECT (i)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a 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 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: 2.3.3.2 | 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)

NOTE – Omitted when no parameter is present.

| | |
|---|---|
| TEST NUMBER: 2.3.3.3 | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774 | |
| TITLE: Inopportune Behaviour; Unexpected Components | |
| SUBTITLE: Return Result Not-Last for Class 2 | |
| PURPOSE: To verify that a rejection can be sent if a Return Result Not-Last component 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 AND COMPONENT 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

| | |
|---|---|
| 1. | Initiate a Class 2 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 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: 2.3.3.3 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN RESULT NOT-LAST component in TSL message from SP B to SP A<br><br>Component type tag: 10100111 (RETURN RESULT NOT-LAST)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Sequence tag: 00110000 (see Note)<br>Sequence length: correct number of octets (see Note)<br><br>Operation code tag: 00000010 (local) or 00000110 (global) (see Note)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long) (see Note)<br>Operation code: x (see Note)<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000010 (RETURN RESULT)<br>Problem code length: 00000001<br>Problem code: 00000001 (return result unexpected)<br><br>NOTE – Omitted when no parameter is present. | |

| | | |
|---|---|---|
| TEST NUMBER: 2.3.3.4 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1/Q.774 | | |
| TITLE: Inopportune Behaviour; Unexpected Components | | |
| SUBTITLE: Return Result Not-Last for Class 4 | | |
| PURPOSE: To verify that a rejection can 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 COMPONENT 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

| | |
|---|---|
| 1. | Initiate a 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 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)

NOTE – Omitted when no parameter is present.

| | |
|---|---|
| TEST NUMBER: 2.3.3.5 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774

TITLE: Inopportune Behaviour; Unexpected Components

SUBTITLE: Return Error for Class 3

PURPOSE: To verify that a rejection can be sent if a Return Error component is received for a Class 3 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 Error component can be generated

| | | |
|---|---|---|
| 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)** | ⟶ | |
| | ⟵ | **RETURN ERROR (i)** |
| *TC-L-REJECT ind.* <============ | | |
| **REJECT (i)** | ⟶ | |

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? |
| 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
Problem code: 00000001 (unexpected return error)

| | |
|---|---|
| TEST NUMBER: 2.3.3.6 | Sheet: 1 of 2 |

REFERENCE: 3.2.1/Q.774

TITLE: Inopportune Behaviour; Unexpected Components

SUBTITLE: Return Error for Class 4

PURPOSE: To verify that a rejection can be sent if a Return Error 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 Error component can be generated

| | | |
|---|---|---|
| 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)** | ⟶ | |
| | ⟵ | **RETURN ERROR (i)** |
| *TC-L-REJECT ind.* <============ | | |
| **REJECT (i)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Initiate a 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 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: 2.3.3.6 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR COMPONENTS WITHIN MESSAGES | |
| Operation code tag: 00000010 (local) or 00000110 (global)<br>Operation code length: correct number of octets (e.g. 00000001 if x is one octet long)<br>Operation code: x (x represents a valid operation code)<br><br>parameters (provided by the TC-User)<br><br>RETURN ERROR component in TSL message from SP B to SP A<br><br>Component type tag: 10100011 (RETURN ERROR)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Error code tag: 00000010 (local) or 00000110 (global)<br>Error code length: correct number of octets (e.g. 00000001 if y is one octet long)<br>Error code: y<br><br>parameters (provided by the TC-User)<br><br>REJECT component in TSL message from SP A to SP B<br><br>Component type tag: 10100100 (REJECT)<br>Component length: correct number of octets<br><br>Invoke ID tag: 00000010<br>Invoke ID length: 00000001<br>Invoke ID: i<br><br>Problem code tag: 10000011 (RETURN ERROR)<br>Problem code length: 00000001<br>Problem code: 00000001 (unexpected return error) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.3.4.1 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.1.2/Q.774 | | |
| TITLE: Inopportune Behaviour; Dialogue Portion | | |
| SUBTITLE: Begin message with APDU AARE | | |
| PURPOSE: To verify that an IUT aborts the transaction on reception of a Begin message containing a dialogue portion that carries an APDU 'AARE' | | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *Detect semantic error* | $\longleftarrow$ | **BEGIN (AARE)** |
| *TR-U-ABORT req* ===========> | | |
| **ABORT (U) (ABRT)** | $\longrightarrow$ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue response. |
| 2. | CHECK A: DOES SP A TRANSMIT THE EXPECTED ABORT MESSAGE WITH APDU 'ABRT'? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Abort message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets

Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H'00118605010101

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Abort tag: 01100100
Dialogue Abort length: 00000011

Abort Source tag: 10000000
Abort source length: 00000001
Abort source: 00000001 (dialogue service provider)

| TEST NUMBER: 2.3.4.1 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Begin<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H'00118605010101<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: 00011001<br>Dialogue PDU<br>Dialogue Response tag: 01100001<br>Dialogue Response length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: any object identifier<br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000000 (Accepted)<br>Result source diagnostic tag: 10100011<br>Result source diag. length: 00000101<br>Dialogue Service User tag: 10100001<br>Dialogue Service User length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service user value: 00000000 (NULL) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.3.4.2 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Inopportune Behaviour; Dialogue Portion | | |
| SUBTITLE: Dialogue confirmation with any APDU other than AARE | | |
| PURPOSE: To verify that an IUT aborts the transaction on reception of a Continue message containing a dialogue portion that carries an APDU 'AARQ' | | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN (AARQ)** | ————————————→ | |
| *Detect semantic error* | ←———————————— | **CONTINUE (AARQ)** |
| *TR-P-ABORT ind.*<br><=========== | | |
| *TR-U-ABORT req.*<br>===========> | | |
| **ABORT (U) (ABRT)** | ————————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion. |
| 2. | Arrange for SP B to confirm the dialogue with a dialogue portion containing an AARQ APDU. |
| 3. | CHECK A: DOES THE DIALOGUE PORTION IN THE ABORT MESSAGE CONTAIN APDU 'ABRT'? |
| 4. | CHECK B: VERIFY THAT THE DIALOGUE AT SP A HAS BEEN TERMINATED. |

| TEST NUMBER: 2.3.4.2 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |

Dialogue portion in Continue message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H'00118605010101

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

Dialogue portion in Abort message

Dialogue portion tag: 01101011
Dialogue portion length: 00010010

External data type in dialogue portion

External type tag: 00101000
External length: 00010000

Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H'00118605010101

Single ASN.1 type tag: 10100000
Single ASN.1 type length: 00000101

Dialogue PDU

Dialogue Abort tag: 01100100
Dialogue Abort length: 00000011

Abort Source tag: 10000000
Abort source length: 00000001
Abort source: 00000001 (dialogue service provider)

| | |
|---|---|
| TEST NUMBER: 2.3.4.3 | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.1/Q.774 | |
| TITLE: Inopportune Behaviour; Dialogue Portion | |
| SUBTITLE: Dialogue confirmation with APDU ABRT | |
| PURPOSE: To verify that an IUT aborts the transaction on reception of a Continue message containing a dialogue portion that carries an ’ABRT’ APDU | |
| 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 (CSL) | | SP B (CSL) |
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN (AARQ)** | ——————————→ | |
| *Detect semantic error* | ←—————————— | **CONTINUE (ABRT)** |
| *TR-P-ABORT ind.*<br><=========== | | |
| *TR-U-ABORT req.*<br>===========> | | |
| **ABORT (U) (ABRT)** | ——————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion. |
| 2. | Arrange for SP B to confirm the dialogue with a dialogue portion containing a ABRT APDU. |
| 3. | CHECK A: DOES THE DIALOGUE PORTION IN THE ABORT MESSAGE CONTAIN APDU ’ABRT’? (refer to test number 2.3.4.2 check table for ABRT). |
| 4. | CHECK B: VERIFY THAT THE DIALOGUE AT SP A HAS BEEN TERMINATED. |

| TEST NUMBER: 2.3.4.3 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Continue message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: 00010010<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: 00010000<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>direct reference: H'00118605010101<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: 00000101<br>Dialogue PDU<br>Dialogue Abort tag: 01100100<br>Dialogue Abort length: 00000011<br>Abort Source tag: 10000000<br>Abort source length: 00000001<br>Abort source: 00000001 (dialogue service provider) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.3.4.4 | | Sheet: 1 of 2 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Inopportune Behaviour; Dialogue Portion | | |
| SUBTITLE: Presence of a Dialogue Portion APDU in the active state. | | |
| PURPOSE: To verify that an IUT aborts the transaction on reception of a Continue message containing a dialogue portion that carries an ’AARE’ APDU | | |
| PRE-TEST CONDITIONS: SP A (TSL) and SP B (TSL) are to be in the idle state and test case 2.1.9.1.2 has to be executed successfully | | |
| CONFIGURATION: 1 | TYPE OF TEST: VAT | TYPE OF SP: SP |

EXPECTED MESSAGE SEQUENCE:

| SP A (CSL) | | SP B (CSL) |
|---|---|---|
| | ⟵ | **BEGIN (AARQ)** |
| *TR-BEGIN ind.* <============ | | |
| *TR-CONTINUE req.* ============> | | |
| **CONTINUE (AARE)** | ⟶ | |
| | ⟵ | **CONTINUE (AARE)** |
| *TR-P-ABORT ind.* <============ | | |
| *TR-U-ABORT req* ============> | | |
| **ABORT (U) (ABRT)** | ⟶ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message with dialogue request to SP A. |
| 2. | Arrange for SP A to confirm the dialogue. |
| 3. | Arrange for SP B to send a Continue msg containing a dialogue portion carrying an AARE APDU. |
| 4. | CHECK A: VERIFY THAT THE IUT AT SP A TERMINATES THE TRANSACTION. |

| TEST NUMBER: 2.3.4.4 | Sheet: 2 of 2 |
|---|---|
| CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION | |
| Dialogue portion in Continue message<br>Dialogue portion tag: 01101011<br>Dialogue portion length: correct number of octets<br>External data type in dialogue portion<br>External type tag: 00101000<br>External length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: 00000111<br>Direct reference: H’00118605010101<br>Single ASN.1 type tag: 10100000<br>Single ASN.1 type length: correct number of octets<br>Dialogue PDU<br>Dialogue Response tag: 01100001<br>Dialogue Response length: correct number of octets<br>Application context tag: 10100001<br>Application context length: correct number of octets<br>Object Identifier tag: 00000110<br>Object Identifier length: correct number of octets<br>Direct reference: same octets as in AARQ<br>Result tag: 10100010<br>Result length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Result value: 00000000 (Accepted)<br>Result source diagnostic tag: 10100011<br>Result source diag. length: 00000101<br>Dialogue Service User tag: 10100001<br>Dialogue Service User length: 00000011<br>INTEGER type tag: 00000010<br>INTEGER length: 00000001<br>Dialogue service user value: 00000000 (NULL) | |

| | | |
|---|---|---|
| TEST NUMBER: 2.3.4.5 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Inopportune Behaviour; Dialogue Portion | | |
| SUBTITLE: Unidirectional message with unexpected abstract syntax | | |
| PURPOSE: To verify that an IUT discards an UNIDIRECTIONAL message containing a dialogue portion that is referring to an unexpected abstract syntax | | |
| 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *Detect error* | $\longleftarrow$ | **UNIDIRECTIONAL (AARQ)** |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Unidirectional message containing a dialogue portion but referring to the structured dialogue abstract syntax. |
| 2. | CHECK A: DOES SP A DISCARD THE MESSAGE? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN MESSAGES

Dialogue portion in Unidirectional message

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H’00118605010101 (structured dialogue abstract syntax)

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

| | | |
|---|---|---|
| TEST NUMBER: 2.3.4.6 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Inopportune Behaviour; Dialogue Portion | | |
| SUBTITLE: Unexpected dialogue portion in Continue message | | |
| PURPOSE: To verify that an IUT aborts the transaction on reception of a Continue message which includes a dialogue portion while no dialogue portion was sent in the Begin 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN** | ————————→ | |
| *Detect error* | ←———————— | **CONTINUE (AARE)** |
| *TR-P-ABORT ind.*<br><=========== | | |
| *TR-U-ABORT req*<br>===========> | | |
| **ABORT (U) (ABRT)** | ————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message (without dialogue portion) |
| 2. | Arrange for SP B to confirm the dialogue with a dialogue portion containing an AARE APDU. (Refer to check table of test number 2.3.4.4.) |
| 3. | CHECK A: DOES THE DIALOGUE PORTION IN THE ABORT MESSAGE CONTAIN APDU 'ABRT'? (Refer to check table of test number 2.3.4.2.) |
| 4. | CHECK B: VERIFY THAT THE DIALOGUE AT SP A HAS BEEN TERMINATED. |

| | | |
|---|---|---|
| TEST NUMBER: 2.3.4.7 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Inopportune Behaviour; Dialogue Portion | | |
| SUBTITLE: Missing dialogue portion in Continue message | | |
| PURPOSE: To verify that an IUT aborts the transaction on reception of a Continue message without dialogue portion while a dialogue portion was sent in the Begin 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 (CSL) | | SP B (CSL) |
|---|---|---|
| *TR-BEGIN req.*<br>===========> | | |
| **BEGIN (AARQ)** | ——————————→ | |
| *Detect error* | ←—————————— | **CONTINUE** |
| *TR-P-ABORT ind.*<br><=========== | | |
| *TR-U-ABORT req*<br>===========> | | |
| **ABORT (U) (ABRT)** | ——————————→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP A to send a Begin message containing a dialogue portion. |
| 2. | Arrange for SP B to confirm the dialogue with a Continue without a dialogue portion. |
| 3. | CHECK A: VERIFY THAT THE DIALOGUE AT SP A HAS BEEN TERMINATED. |

| | | |
|---|---|---|
| TEST NUMBER: 2.3.4.8 | | Sheet: 1 of 1 |
| REFERENCE: 3.2.2.1/Q.774 | | |
| TITLE: Inopportune Behaviour; Dialogue Portion | | |
| SUBTITLE: Begin message with unexpected abstract syntax | | |
| PURPOSE: To verify that an IUT aborts the transaction on reception of a Begin message containing a dialogue portion referring to an unexpected abstract syntax | | |
| 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 (CSL) | | SP B (CSL) |
| *Detect error* | ←―――――――――――――― | **BEGIN (AUDT)** |
| *TR-U-ABORT req* ===========> | | |
| **ABORT (U) (ABRT)** | ――――――――――――――→ | |

TEST DESCRIPTION

| | |
|---|---|
| 1. | Arrange for SP B to send a Begin message containing a dialogue portion referring to an unknown abstract syntax. |
| 2. | CHECK A: DOES SP A TERMINATE THE TRANSACTION? |

CHECK TABLE FOR INFORMATION ELEMENTS WITHIN DIALOGUE PORTION

Dialogue portion in Begin

Dialogue portion tag: 01101011
Dialogue portion length: correct number of octets

External data type in dialogue portion

External type tag: 00101000
External length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: 00000111
Direct reference: H’00118605010201 ( unstructured abstract syntax )

Single ASN.1 type tag: 10100000
Single ASN.1 type length: correct number of octets

Dialogue PDU

Dialogue Request tag: 01100000
Dialogue Request length: correct number of octets

Application context tag: 10100001
Application context length: correct number of octets
Object Identifier tag: 00000110
Object Identifier length: correct number of octets
Direct reference: any object identifier

## ITU-T RECOMMENDATIONS SERIES