[MC-DTCXA]:
MSDTC Connection Manager:
OleTx XA Protocol Specification

This protocol documentation is intended for use in conjunction with publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

Revision Summary

Date	Revision History	Revision Class	Comments
08/10/2007	0.1	Major	Initial Availability
09/28/2007	0.2	Minor	Updated the technical content.
10/23/2007	0.2.1	Editorial	Revised and edited the technical content.
11/30/2007	0.2.2	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
01/25/2008	0.2.3	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

This document specifies the extensions to the MSDTC Connection Manager: OleTx Transaction Protocol [MS-DTCO] to support XA-compliant software components, as specified in [XOPEN-DTP], in an **OleTx** distributed **transaction** processing environment. It specifies the syntax and semantics of the new protocol messages. The document builds upon and relies heavily upon the MSDTC Connection Manager: OleTx Transaction Protocol specification [MS-DTCO], and readers must be familiar with its terms and concepts.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Application
ASCII
Atomic Transaction
Connection
Globally Unique Identifier (GUID)
Message Tag (MTAG)
Phase Two
Recovery
Subordinate Transaction Manager
Superior Transaction Manager
Transaction
Transaction Identifier
Transaction Manager (TM)
Two-Phase Commit

The following terms are defined in [MS-DTCO]:

Enlistment
Facet
In Doubt Outcome
OleTx
Outcome
Participant
Phase One Enlistment
Resource Manager (RM)
Transient Failure
Work

The following terms are specific to this document:

Branch: See **XA Transaction Branch**.

Child Branch: The second or later **XA Transaction Branch** created on an XA Subordinate Transaction Manager Facet for a given **XA Global Transaction Identifier (GTRID)** when using **tight coupling**.

Loose Coupling: A scheme for mapping **XA Transaction Branches** to **atomic transactions**. Each **loosely coupled XA Transaction Branch** is treated as operating under a different **atomic transaction** by the XA Subordinate Transaction Manager Facet.

One pipe: A model of communication between an XA Resource Manager Bridge, an **XA Transaction Manager**, and an **XA Resource Manager**. For more information, see the description in 1.3.1.2.

One-Pipe XA Resource Manager: An **XA Resource Manager** that uses the **one-pipe** model to communicate with an **XA Transaction Manager**.

Parent Branch: The first **XA Transaction Branch** created on an XA Subordinate Transaction Manager Facet for a given **XA Global Transaction Identifier** when using **tight coupling**.

Resource Manager Cookie: An identifier used to uniquely identify an **XA Resource Manager** Proxy object between calls to XA Resource Manager Bridge high-level events.

Tight Coupling: A scheme for mapping **XA Transaction Branches** to **atomic transactions**. All **tightly coupled XA Transaction Branches** with the same **XA Global Transaction Identifier** are treated as operating under one **atomic transaction** by the XA Subordinate Transaction Manager Facet.

Two Pipe: This is a model of communication between an XA Resource Manager Bridge, an **XA Transaction Manager** and an **XA Resource Manager**. For more information, see the description in 1.3.1.2.

Two-Pipe XA Resource Manager: An **XA Resource Manager** that uses the **two-pipe** model to communicate with an **XA Transaction Manager**.

XA Branch Identifier: The **globally unique identifier (GUID)** used by an XA Resource Manager Bridge to generate the **XA Branch Qualifier (BQUAL)** of an **XA Transaction Branch Identifier (XID)**. This **GUID** uniquely identifies the **XA Transaction Branch** within an XA Resource Manager Bridge Facet and **XA Resource Manager**.

XA Branch Qualifier (BQUAL): A field of an **XID** that uniquely identifies an **XA Transaction Branch** within a **transaction**. For more information, see [XOPEN-DTP].

XA Format Identifier: A format identifier for an **XID**. For more information, see [XOPEN-DTP].

XA Global Transaction Identifier (GTRID): A field of an **XID** that uniquely identifies a **transaction**. For more information, see [XOPEN-DTP].

XA Protocol: The protocol specified in [XOPEN-DTP].

XA Resource Manager: A **resource manager** that uses the protocol specified in [XOPEN-DTP] to communicate with an **XA Transaction Manager**.

XA Resource Manager Bridge: A software component that allows an application to enlist an XA Resource Manager in an OleTx Transaction.

XA Resource Manager Bridge Facet: A software component that allows a Transaction Manager to communicate with an XA Resource Manager Bridge.

XA Resource Manager Instance Identifier: An identifier used to uniquely identify an instance of an **XA Resource Manager**. It does not persist through failure or software restart.

XA Superior Transaction Manager Identifier (Resource Manager Recovery GUID): A **GUID** used to uniquely identify an XA Superior Transaction Manager to an XA Subordinate Transaction Manager Facet. This identifier must persist through **transient failure** and **recovery**.

XA Transaction Branch: Represents a single Unit of Work done under a **transaction**. For more information, see [XOPEN-DTP].

XA Transaction Branch Identifier (XID): An identifier for an **XA Transaction Branch**.

XA Transaction Manager: A **Superior Transaction Manager** that uses the protocol specified in [XOPEN-DTP] to communicate with **XA Resource Managers**.

XA Transaction Manager Identifier: A **GUID** used by an XA Resource Manager Bridge to generate the **BQUAL** of an **XID**. This **GUID** uniquely identifies the XA Resource Manager Bridge Facet that the **XID** is associated with.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[ISO-8859-1] International Organization for Standardization, "Information Technology -- 8-Bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1", ISO/IEC 8859-1, 1998, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=28245&ICS1=35&ICS2=40&ICS3=

Note There is a charge to download the specification.

[MS-CMP] Microsoft Corporation, "MSDTC Connection Manager: OleTx Multiplexing Protocol Specification", July 2007.

[MS-CMPO] Microsoft Corporation, "MSDTC Connection Manager: OleTx Transports Protocol Specification", July 2007.

[MS-DTCO] Microsoft Corporation, "MSDTC Connection Manager: OleTx Transaction Protocol Specification", July 2007.

[MS-DTYP] Microsoft Corporation, "Windows Data Types", January 2007.

[MS-ERREF] Microsoft Corporation, "Windows Error Codes", January 2007.

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[XOPEN-DTP] The Open Group, "Distributed Transaction Processing: The XA Specification", December 1991, http://www.opengroup.org/bookstore/catalog/c193.htm

1.2.2 Informative References

There are no informative references for this document.

1.3 Protocol Overview (Synopsis)

In a distributed transaction, typically three types of software components are involved:

- An application program (AP) defines transaction boundaries and specifies actions that constitute a transaction.
- **Resource managers (RMs)**, such as databases or file access systems, provide access to shared resources.
- A separate component called a **transaction manager (TM)** assigns identifiers to transactions, monitors their progress, and takes responsibility for transaction completion and for failure **recovery**.

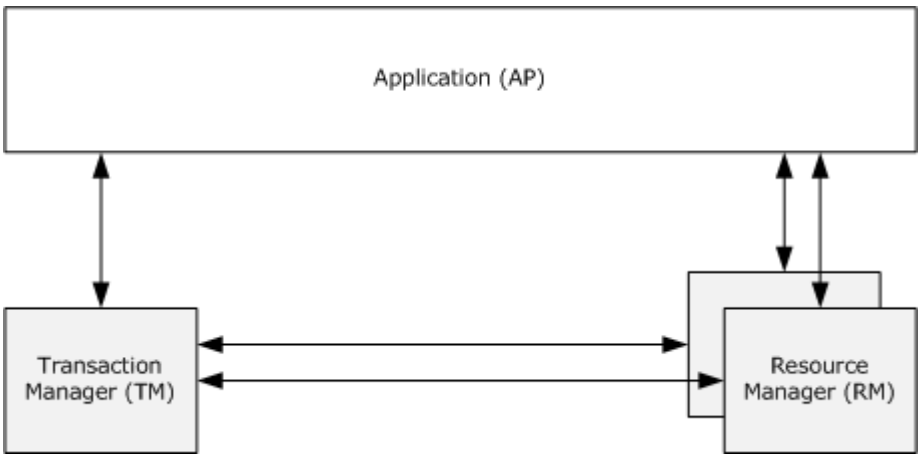


Figure 1: Software components of a typical distributed transaction

The MSDTC Connection Manager: OleTx Transaction Protocol specification [MS-DTCO] specifies a comprehensive distributed transaction protocol (OleTx). X/Open specifies a distributed transaction processing model, and a bidirectional XA interface [XOPEN-DTP] between a TM and an RM. There are certain differences (such as a difference in the syntax and semantics of the **transaction identifiers**) between the OleTx distributed transaction processing model, and the XA distributed transaction processing model. The protocol extensions specified in this document bridge those differences, and in particular solve the following problems:

- Enable OleTx Resource Managers to participate in transactions coordinated by **XA Transaction Managers**. This scenario is further discussed in 1.3.1.1.
- Enable **XA Resource Managers** to participate in transactions coordinated by OleTx Transaction Managers. This scenario is further discussed in 1.3.1.2.

1.3.1 Scenarios

1.3.1.1 OleTx Resource Managers Enlisting with XA Transaction Managers

The communications between an XA Superior Transaction Manager and an XA Subordinate Transaction Manager Facet specified in this document enable OleTx Resource Managers to participate in transactions coordinated by XA Transaction Managers. A **subordinate transaction manager** software component with an XA Subordinate Transaction Manager Facet facilitates this scenario. The following diagram shows components involved in this usage scenario.

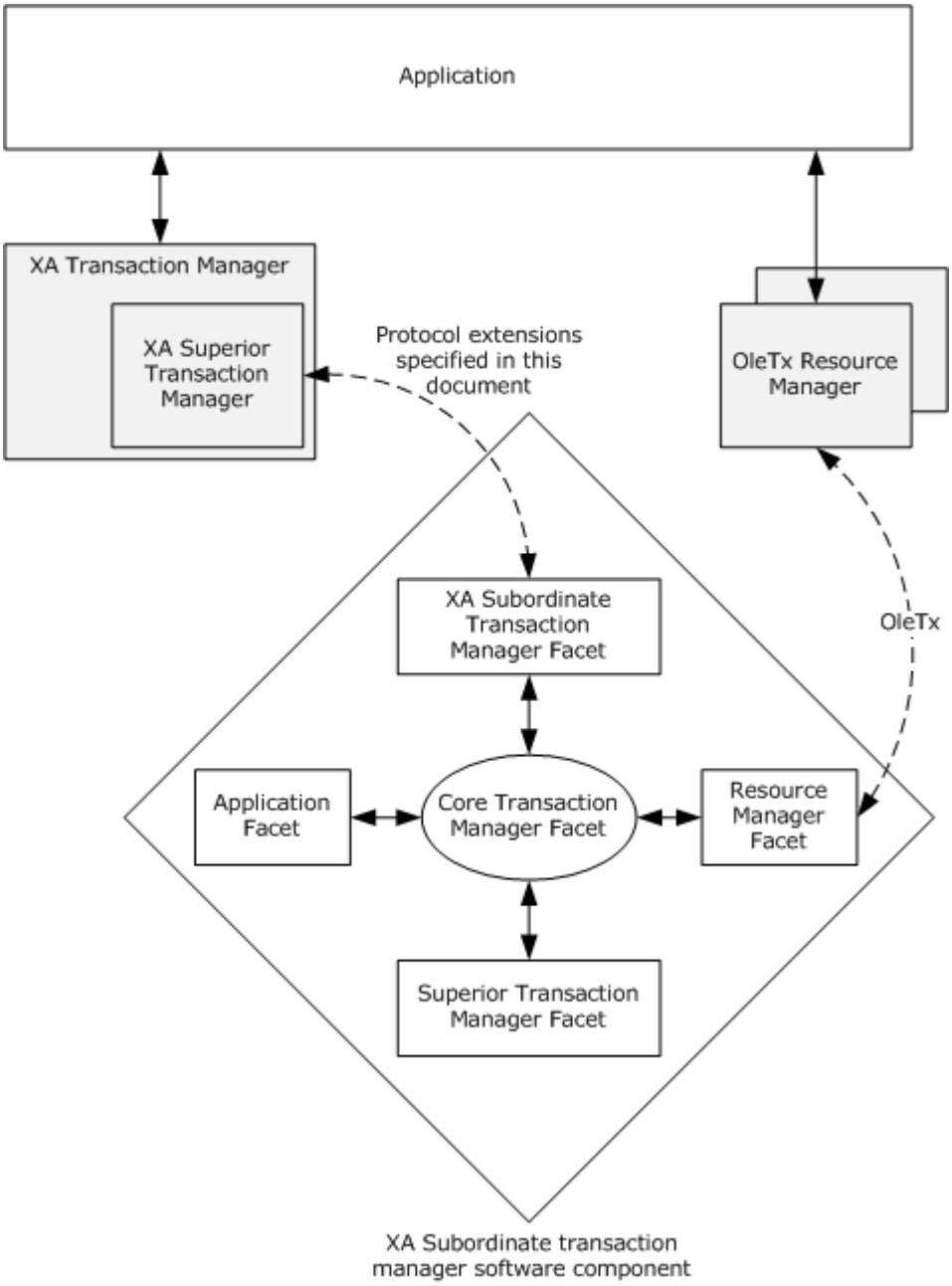


Figure 2: OleTx Resource Managers enlisting with XA Transaction Managers' topology

The following sections illustrate the interactions that take place between these components in a common scenario drawn from each of the following areas of a distributed transaction processing.

- Transaction **enlistment** and completion
- Transaction recovery

1.3.1.1.1 Transaction Enlistment and Completion

The following sequence diagram shows the messages that are exchanged between the various software components involved in the usage scenario when an OleTx Resource Manager is enlisted in a transaction coordinated by an XA Transaction Manager, and the transaction is committed.

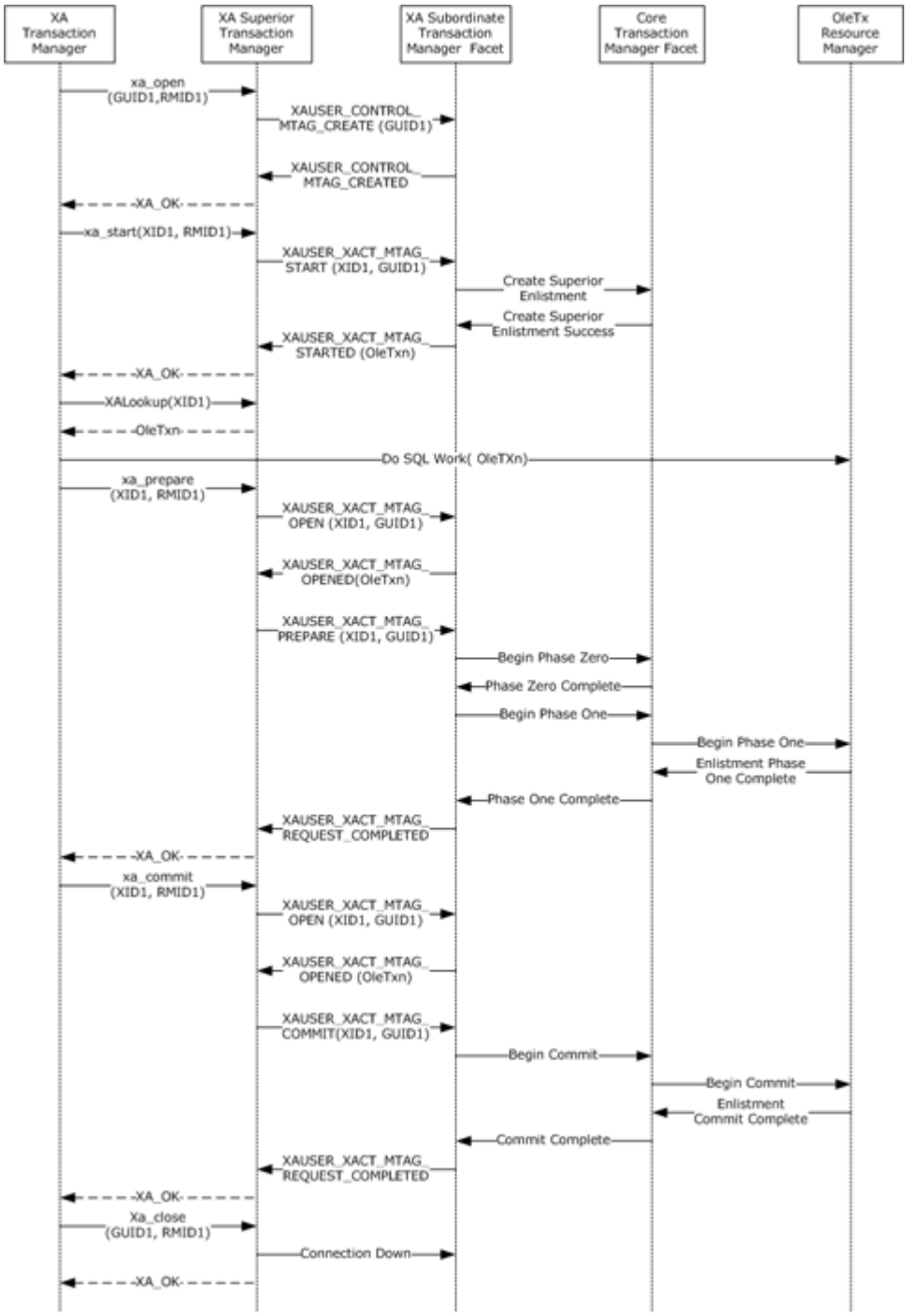


Figure 3: XA Superior Transaction enlistment and completion

All the exchanges depicted are notional, and are not intended to provide an accurate representation of any concrete protocol. The protocols involved are specified as follows:

- The protocol between the XA Superior Transaction Manager and the XA Subordinate Transaction Manager Facet is specified by this document in 2 and 3.

The protocol between the XA Subordinate Transaction Manager Facet and the Core Transaction Manager Facet is specified in [MS-DTCO].

- The transaction protocol between the XA Transaction Manager and Resource Managers is OleTx. The data access protocol is resource managers-dependent.
- The interface between the XA Transaction Manager and the XA Superior Transaction Manager is specified in [XOPEN-DTP], with the exception of the implementation-specific XA Lookup event.
- The protocol between the OleTx Resource Manager and the Core Transaction Manager Facet is specified in [MS-DTCO].
- The protocol between the **application** and the OleTx Resource Manager is implementation-specific.

To enlist the OleTx Resource Managers, the XA Transaction Manager first calls xa_start on the XA Superior Transaction Manager, passing in an **XID**. This will cause an OleTx transaction to be created in the XA Subordinate Transaction Manager Facet, which will be passed back to the XA Superior Transaction Manager. The XA Transaction Manager then triggers the XA Lookup event on the XA Superior Transaction Manager, passing in the XID, which will return the OleTx transaction.

The OleTx Resource Managers are then enlisted in the OleTx transaction as specified in [MS-DTCO] section 3.5.

The XA Transaction Manager then follows the **Two-Phase Commit Protocol**.

1.3.1.1.2 Transaction Recovery

The following sequence diagram shows the messages that are exchanged between the various software components involved in the usage scenario when an XA Transaction Manager comes up after a crash, and performs recovery for the transactions that it has not yet completed.

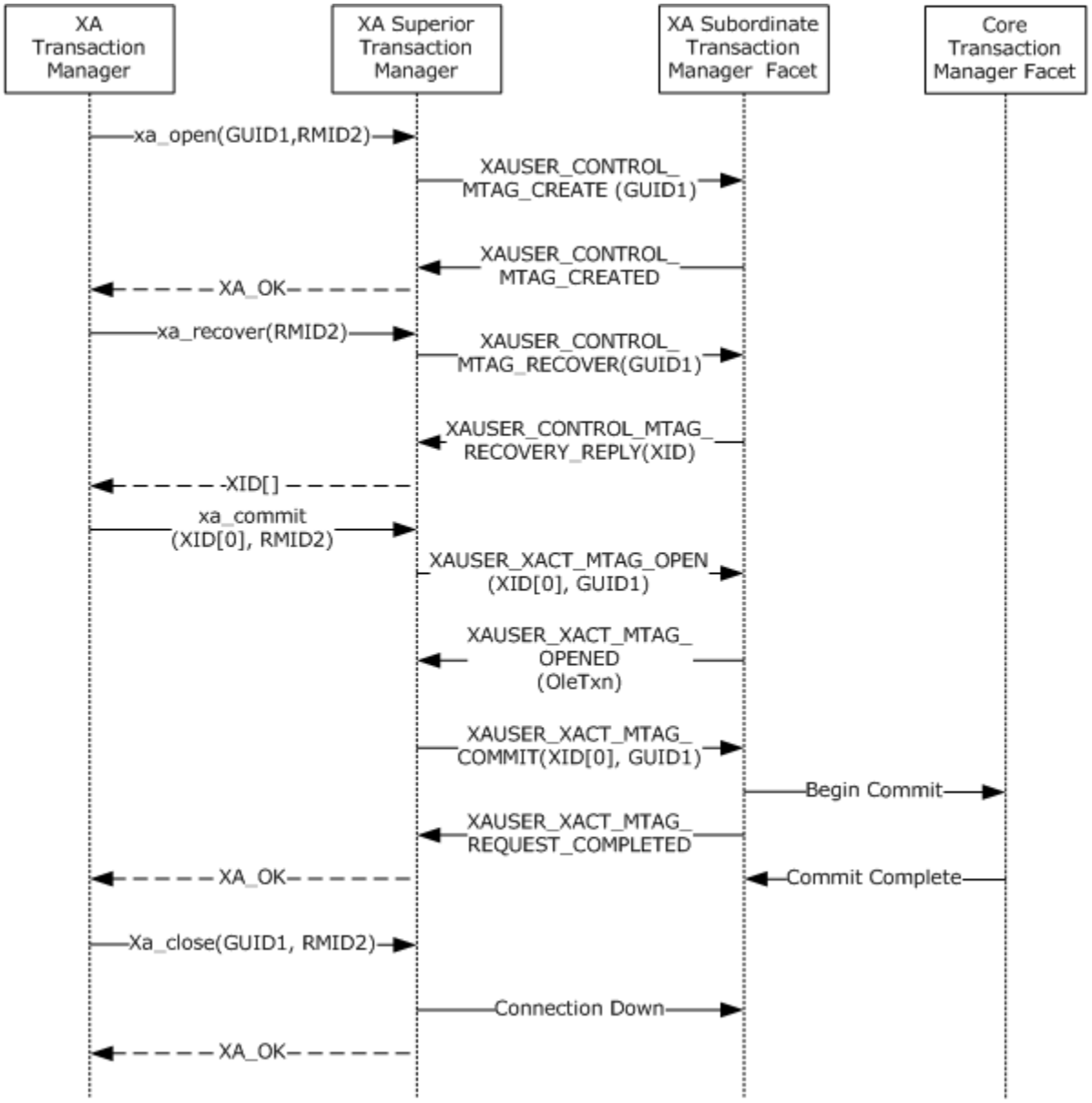


Figure 4: XA Superior Transaction recovery

All the exchanges depicted are notional, and are not intended to provide an accurate representation of any concrete protocol. The protocols involved are specified as follows:

- The protocol between the XA Superior Transaction Manager and the XA Subordinate Transaction Manager Facet is specified by this document in section 2 and section 3.
- The protocol between the XA Subordinate Transaction Manager Facet and the Core Transaction Manager Facet is specified in [MS-DTCO].
- The interface between the XA Transaction Manager and the XA Superior Transaction Manager is specified in [XOPEN-DTP], with the addition of the implementation-specific XA Lookup event.

One of the differences between the OleTx model and the XA model, as far as the recovery process is concerned, is that in the OleTx model the recovery process is initiated by a resource manager that re-enlists, whereas in case of the XA model, the recovery process is initiated by an XA Transaction Manager.

The XA Transaction Manager requests a list of all those transactions in need of recovery, and makes calls on the XA Superior Transaction Manager to communicate their results using the protocol specified in [XOPEN-DTP].

1.3.1.2 XA Resource Managers Enlisting with OleTx Transaction Managers

The communications between an XA Resource Manager Bridge and an XA Resource Manager Bridge Facet specified in this document enable XA Resource Managers to participate in transactions coordinated by OleTx Transaction Managers. A **Superior Transaction Manager** software component with an XA Resource Manager Bridge Facet facilitates this scenario. The following figure shows the components of the OleTx Transaction Managers topology.

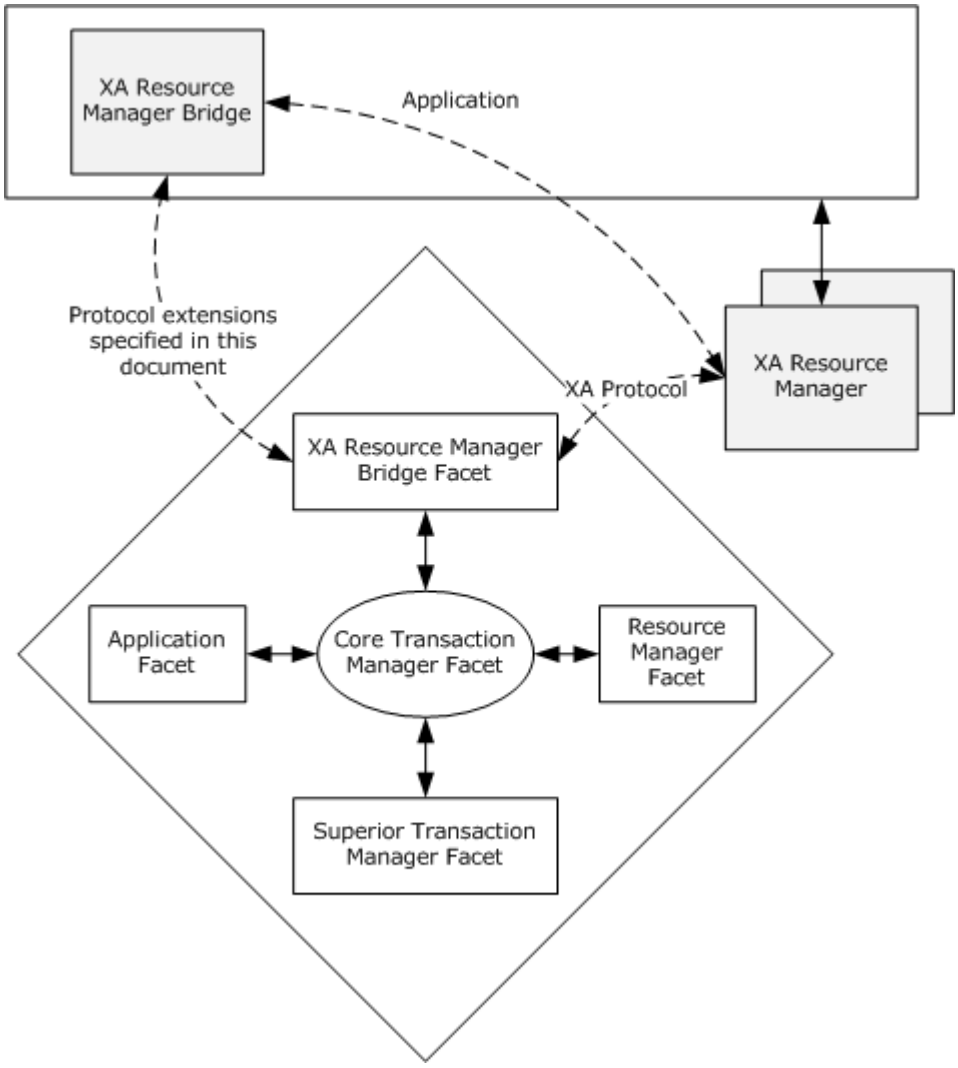


Figure 5: XA Resource Managers enlisting with OleTx Transaction Managers topology

The following sections illustrate the interactions that take place between these components in a common scenario drawn from each of the following areas of a distributed transaction processing.

- Transaction recovery
- XA Resource Manager registration and unregistration
- Transaction enlistment and completion

The protocol defined in this document supports two models to support the "XA Resource Managers enlisting with OleTx Transaction Managers" scenario:

- **Two-pipe** model
- **One-pipe** model

Two pipe is a model of communication between an XA Resource Manager Bridge, an XA Transaction Manager, and an XA Resource Manager, where an XA Transaction Manager makes the XA Protocol calls associated with Two-Phase Commit and Recovery to an XA Resource Manager.

One pipe is another model of communication between an XA Resource Manager Bridge, an XA Transaction Manager, and an XA Resource Manager. In this model, when an XA Resource Manager is registered with the XA Resource Manager Bridge, the XA Resource Manager Bridge registers an OleTx Resource Manager as specified in [MS-DTCO] section 3.5. When a request is made to enlist the XA Resource Manager in an OleTx transaction, the OleTx Resource Manager is enlisted in the OleTx transaction. The XA Resource Manager Bridge makes the necessary Two-Phase Commit calls of the protocol defined by [XOPEN-DTP] to the XA Resource Manager. The only XA Protocol calls made from the XA Transaction Manager to the XA Resource Manager are those associated with recovery.

In both the models, the interactions that take place for transaction recovery between various software components are same. XA Resource Manager registration, unregistration, transaction enlistment, and completion require different interactions in the two models. The following subsections discuss interactions involved for transaction recovery, XA Resource Manager registration, unregistration, and transaction enlistment and completion in two-pipe and one-pipe models.

1.3.1.2.1 Transaction Recovery

The **atomicity** property of a transaction guarantees that all **participants** in the transaction will receive the same **outcome**. In order to honor this guarantee, transaction managers have to be capable of recovering from **transient failures**.

After a transient failure, the transaction manager reloads the xa_switches of each of the registered XA Resource Managers and polls each one for incomplete transactions. It then proceeds to inform the XA Resource Managers of the outcomes of the transactions.

The following sequence diagram shows the messages that are exchanged between the various software components involved when recovering from a transient failure.

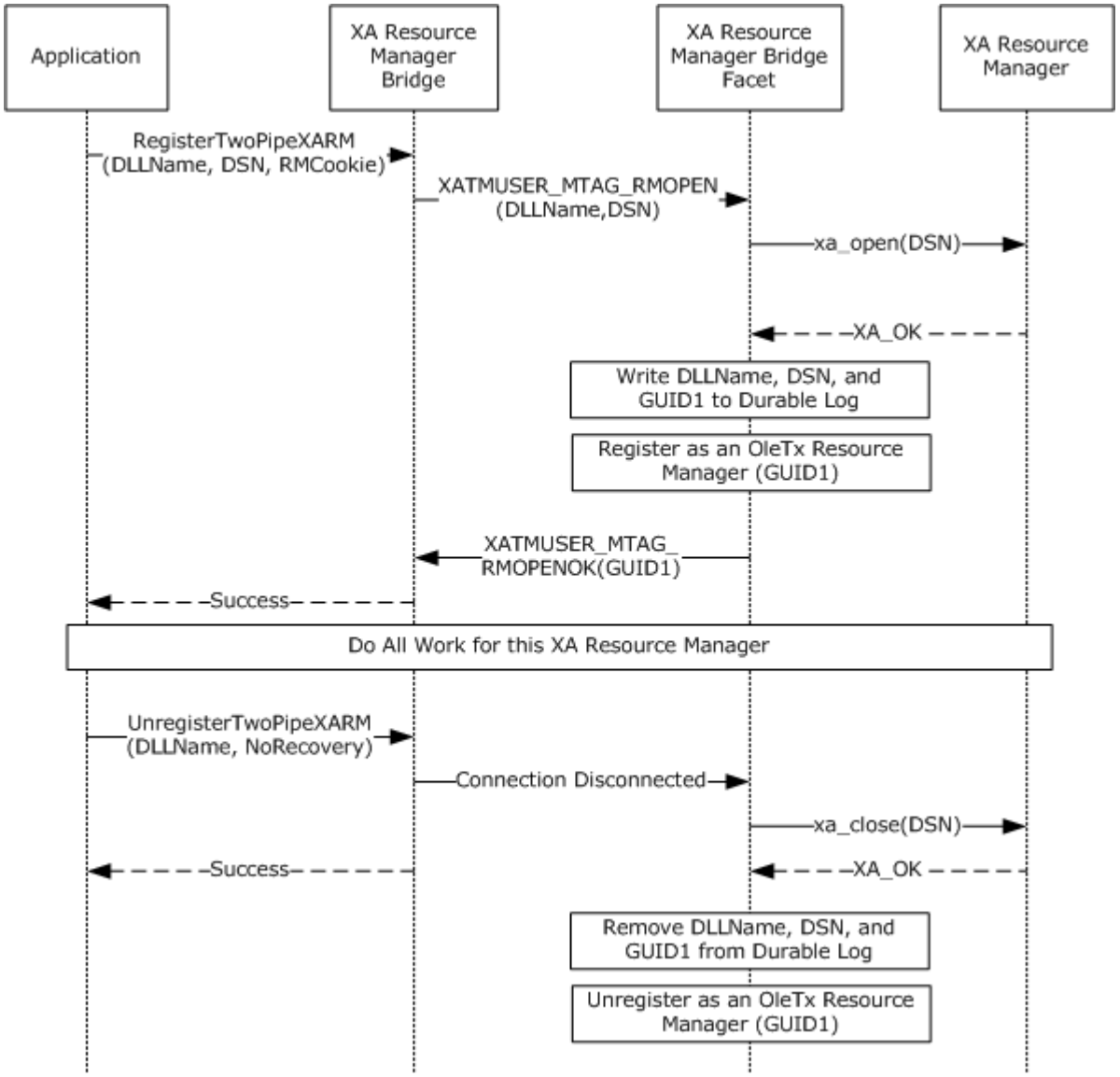


Figure 6: XA Resource Manager Transaction enlistment and completion

All the exchanges depicted are notional, and are not intended to provide an accurate representation of any concrete protocol. The protocols involved are specified as follows:

- The protocol between the XA Resource Manager Bridge Facet and the XA Resource Manager is specified in [XOPEN-DTP].
- The protocol between the XA Resource Manager Bridge Facet and the Core Transaction Manager Facet is specified in [MS-DTCO].
- The intent of these interactions is similar to that of the Recovery protocol between a Transaction Manager and a Resource Manager as described in [MS-DTCO] section 1.3.4. However, the XA Protocol specified in [XOPEN-DTP] specifies that the Transaction Manager is responsible for initiating recovery rather than the Resource Manager.

1.3.1.2.2 Two-Pipe Model

In this model, after an XA Resource Manager is enlisted in an OleTx transaction, the XA Resource Manager Bridge Facet makes the necessary XA Protocol calls for the Two-Phase Commit Protocol when communicating with the XA Resource Manager.

If a transient failure occurs and recovery is necessary, the XA Resource Manager Bridge Facet drives the recovery process with all the registered XA Resource Managers.

1.3.1.2.2.1 XA Resource Manager Registration and Unregistration

The following sequence diagram shows the messages that are exchanged between the various software components involved in the usage scenario when a **Two-Pipe XA Resource Manager** is registered with an OleTx Transaction Manager.

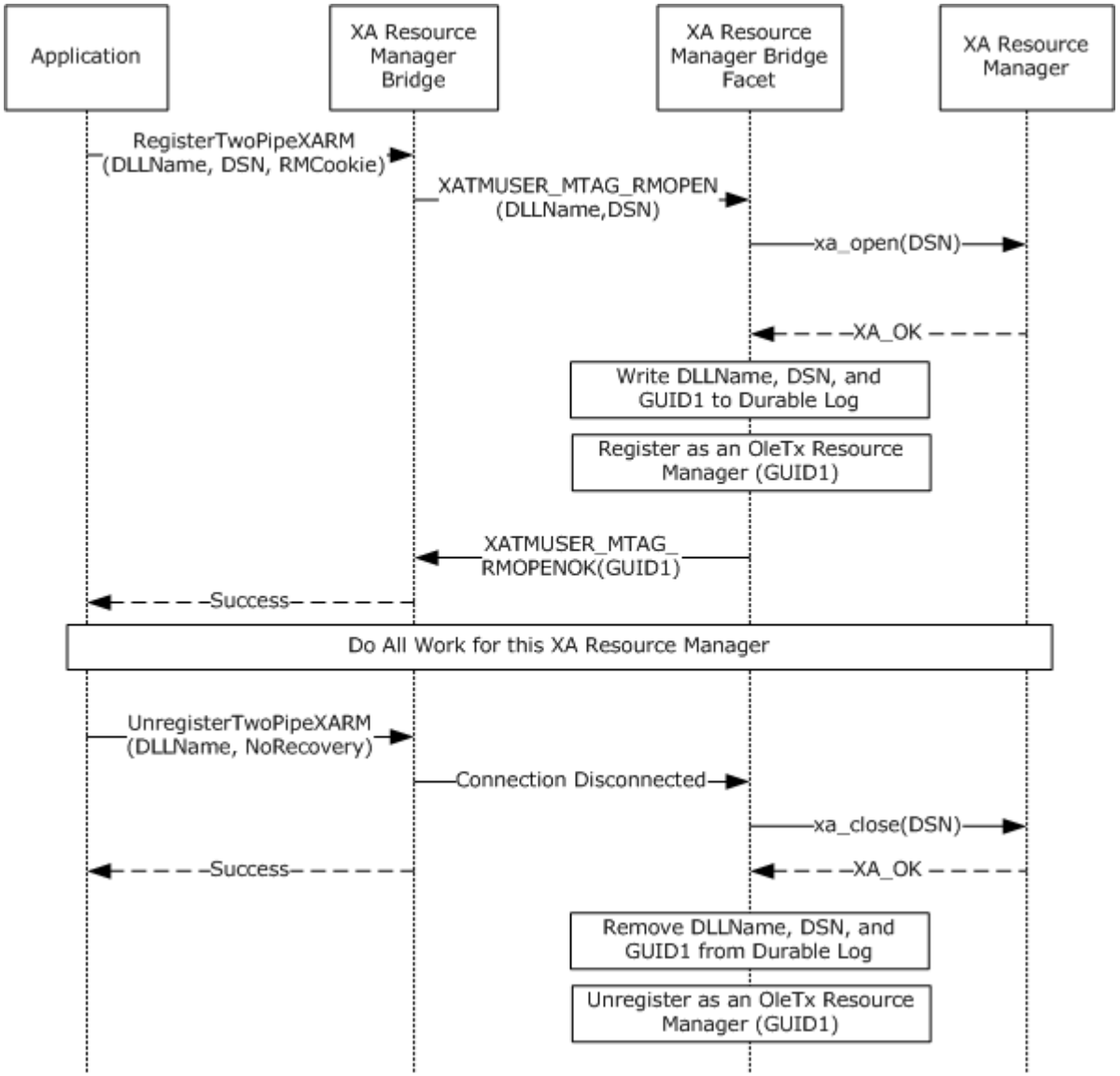


Figure 7: Two-pipe XA Resource Manager registration and unregistration

All the exchanges depicted are notional, and are not intended to provide an accurate representation of any concrete protocol. The protocols involved are specified as follows:

- The protocol between the XA Resource Manager Bridge and the XA Resource Manager Bridge Facet is specified by this document in 2 and section 3.
- The interface between the XA Resource Manager Bridge Facet and the XA Resource Manager is specified in [XOPEN-DTP].
- The protocol between the application and the XA Resource Manager Bridge is implementation-specific.

To register an XA Resource Manager, an application passes the XA Resource Manager Bridge a DLL Name, a Data Source Name, and an **Resource Manager cookie**. The DLL name is used by the XA Resource Manager Bridge Facet to load the xa_switch of the XA Resource Manager. The Data Source Name, an ASCII string, is passed to the xa_open and xa_close calls made on this XA Resource Manager. The RM cookie is used by the application to identify the XA Resource Manager in future calls.

The XA Resource Manager Bridge passes these parameters to the XA Resource Manager Bridge Facet, which loads the xa_switch of the XA Resource Manager and calls xa_open on it. If this succeeds it will write the DLL Name, the Data Source Name, and a **GUID** generated to uniquely identify the XA Resource Manager to a durable log to allow for recovery in case of a transient failure. The XA Resource Manager Bridge Facet then passes the GUID back to the XA Resource Manager Bridge.

The XA Resource Manager Bridge stores the GUID, indexed by the RM cookie, and returns success.

When all **Work** is complete for the XA Resource Manager, it is unregistered by passing the RM cookie to the XA Resource Manager Bridge, which calls the XA Resource Manager Bridge Facet on the same MSDTC Connection Manager: OleTx Multiplexing Protocol (as specified in [MS-CMP]) **connection** used to register the XA Resource Manager and requests that the XA Resource Manager be unregistered.

The XA Resource Manager Bridge Facet removes the entry for the XA Resource Manager from the durable log and returns success.

1.3.1.2.2.2 Transaction Enlistment and Completion

The following sequence diagram shows the messages that are exchanged between the various software components involved in the usage scenario when a Two-Pipe XA Resource Manager is enlisted in an OleTx transaction.

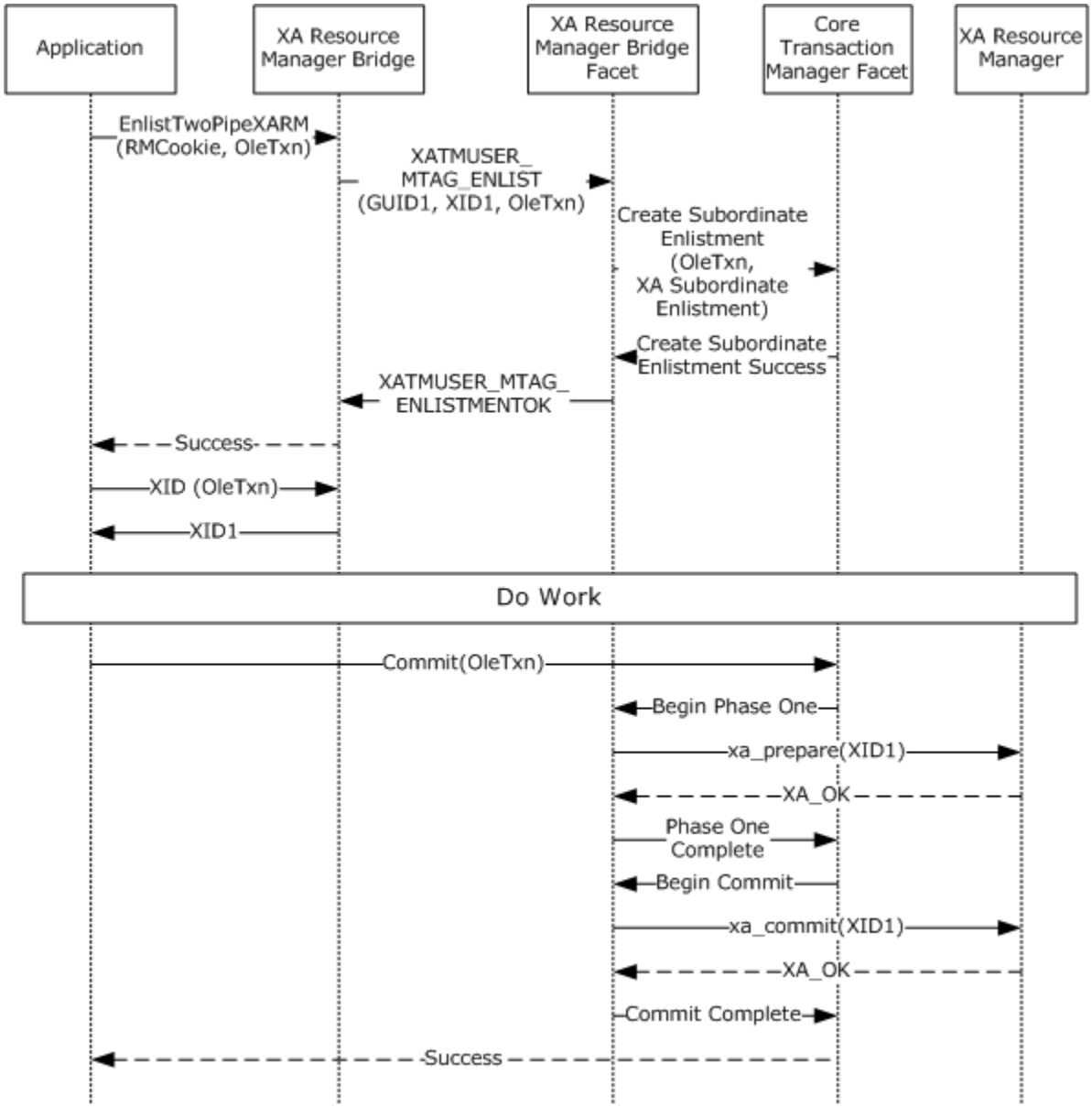


Figure 8: Two-pipe XA Resource Manager Transaction enlistment and completion

All the exchanges depicted are notional, and are not intended to provide an accurate representation of any concrete protocol. The protocols involved are specified as follows:

- The protocol between the XA Resource Manager Bridge and the XA Resource Manager Bridge Facet is specified by this document in section 2 and section 3.
- The protocol between the XA Resource Manager Bridge Facet and the Core Transaction Manager Facet is specified in [MS-DTCO].
- The protocol between the Application and the XA Resource Manager Bridge is implementation-specific.

- The process of enlisting an XA Resource Manager in an OleTx transaction is very similar to enlisting an OleTx Resource Manager in an OleTx transaction.
- After the XA Resource Manager is enlisted in the OleTx transaction, one calls Create XID on the XA Resource Manager Bridge, passing in the OleTxtransaction and receiving a corresponding XID. This XID is then used to perform Work on the XA Resource Manager.
- During the Two-Phase Commit Protocol, when the Subordinate Enlistment created by the XA Resource Manager Bridge Facet receives a prepare request, it constructs the XID associated with the OleTx transaction and calls xa_prepare on the xa_switch of the enlisted XA Resource Manager, passing in the XID.
- When the subordinate enlistment created by the XA Resource Manager Bridge Facet receives a commit request, it calls xa_commit on the xa_switch of the enlisted XA Resource Manager, passing in the XID.

1.3.1.2.3 One-Pipe Model

In this model, after an XA Resource Manager is enlisted in an OleTx transaction, an implementation-specific enlistment in the OleTx transaction makes the calls necessary for the Two-Phase Commit Protocol, using the protocol specified in [XOPEN-DTP], to the XA Resource Manager.

If a transient failure occurs and recovery is necessary, the XA Resource Manager Bridge Facet drives the recovery process with all registered XA Resource Managers.

1.3.1.2.3.1 XA Resource Manager Registration and Unregistration

The following sequence diagram shows the messages that are exchanged between the various software components involved in the usage scenario when a **One-Pipe XA Resource Manager** is registered with an OleTx Transaction Manager.

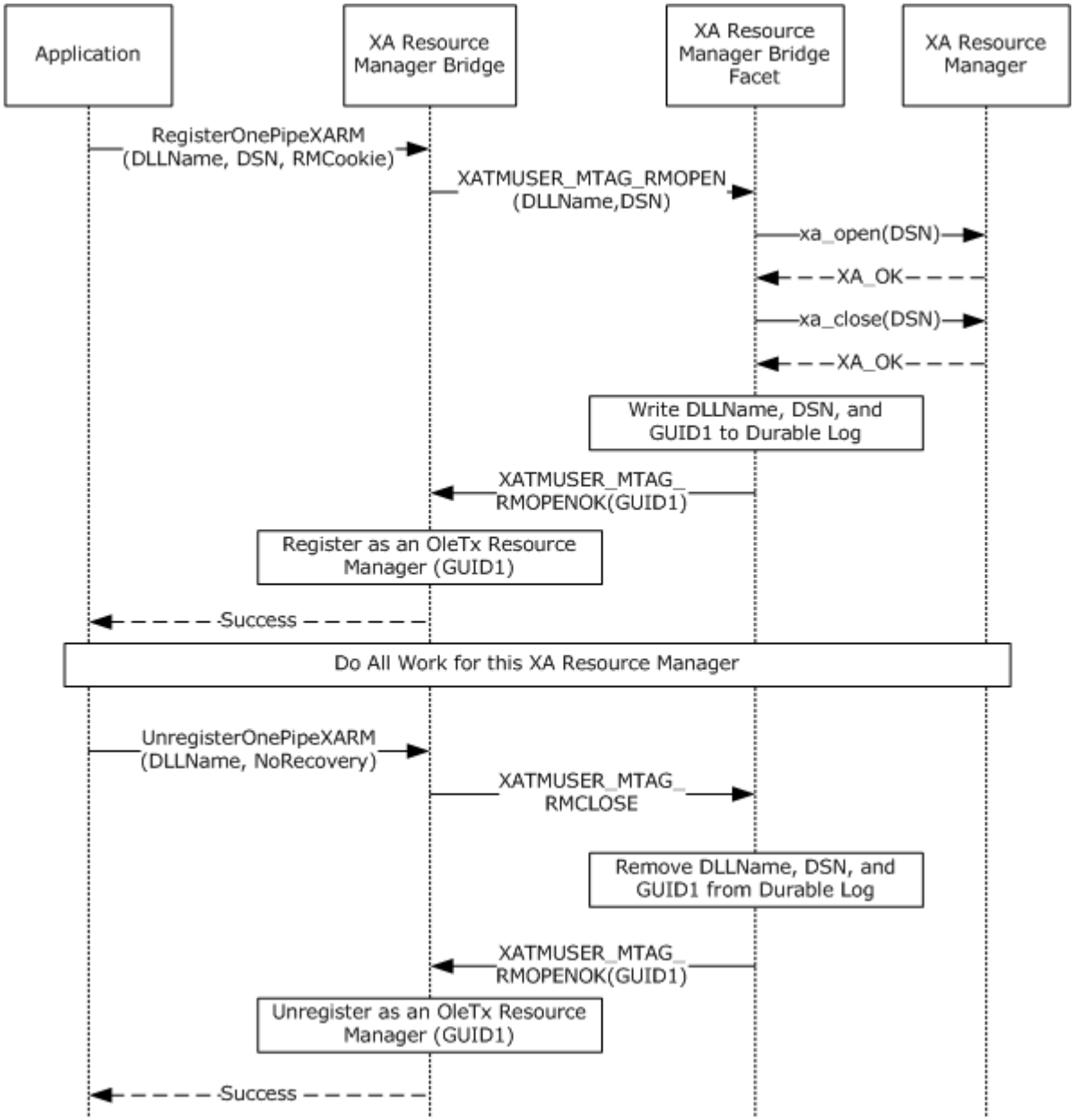


Figure 9: One-pipe XA Resource Manager registration and unregistration

All the exchanges depicted are notional, and are not intended to provide an accurate representation of any concrete protocol. The protocols involved are specified as follows:

- The protocol between the XA Resource Manager Bridge and the XA Resource Manager Bridge Facet is specified by this document in section 2 and section 3.
- The interface between the XA Resource Manager Bridge Facet and the XA Resource Manager is specified in [XOPEN-DTP].

- The protocol between the XA Resource Manager Bridge and the Core Transaction Manager Facet is specified in [XOPEN-DTP].
- The protocol between the application and the XA Resource Manager Bridge is implementation-specific.

To register an XA Resource Manager, an application passes the XA Resource Manager Bridge a DLL Name, a Data Source Name, and an RM cookie. The DLL name is used by the XA Resource Manager Bridge Facet to load the xa_switch of the XA Resource Manager. The Data Source Name, an ASCII string, is passed to the xa_open and xa_close calls made on this XA Resource Manager. The RM cookie is used by the application to identify the XA Resource Manager in future calls.

The XA Resource Manager Bridge passes these parameters to the XA Resource Manager Bridge Facet, which loads the xa_switch of the XA Resource Manager and calls xa_open and then xa_close on it. If this succeeds it will write the DLL Name, the Data Source Name, and a GUID generated to uniquely identify the XA Resource Manager to a durable log to allow for recovery in case of a transient failure. The XA Resource Manager Bridge Facet then passes the GUID back to the XA Resource Manager Bridge.

The XA Resource Manager Bridge uses the GUID to register an OleTx Resource Manager, then stores the GUID, indexed by the RM cookie, and returns success.

When all Work is complete for the XA Resource Manager, it is unregistered by passing the RM cookie to the XA Resource Manager Bridge along with a flag to indicate whether recovery should be initiated on the XA Resource Manager.

If no recovery is necessary, an RMCLOSE request is sent on the same CMP connection (as specified in [MS-CMP]) used to register the XA Resource Manager to unregister XA Resource Manager.

The XA Resource Manager Bridge Facet removes the entry for the XA Resource Manager from the Durable log and returns success.

If recovery is necessary, the CMP connection (as specified in [MS-CMP]) used to register the XA Resource Manager is brought down, which will trigger recovery.

1.3.1.2.3.2 Transaction Enlistment and Completion

The following sequence diagram shows the messages that are exchanged between the various software components involved in the usage scenario when a One-Pipe XA Resource Manager is enlisted in an OleTx transaction.

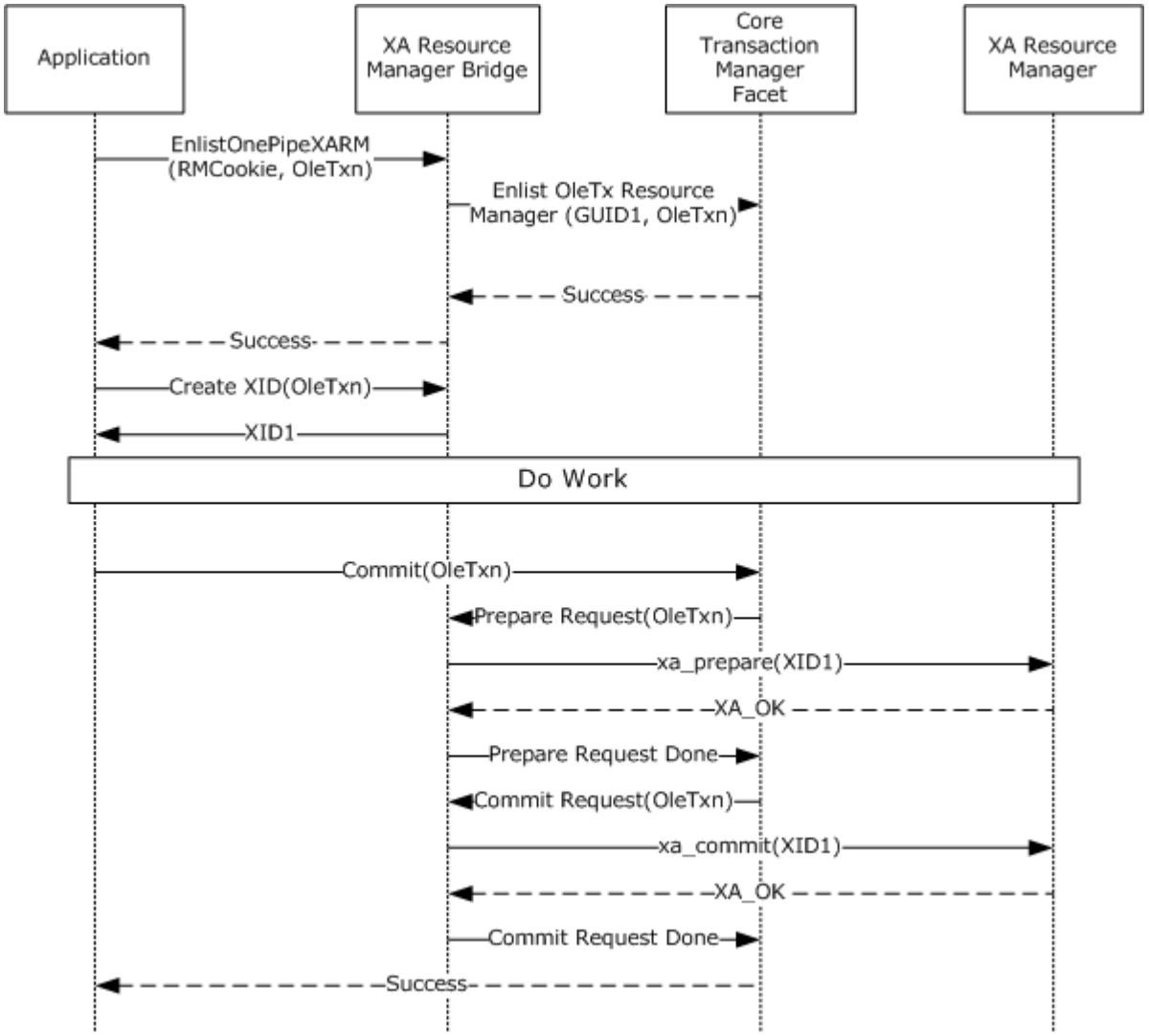


Figure 10: One-pipe XA Resource Manager transaction enlistment and completion

All the exchanges depicted are notional, and are not intended to provide an accurate representation of any concrete protocol. The protocols involved are specified as follows:

- The protocol between the XA Resource Manager Bridge and the Core Transaction Manager Facet is specified in [MS-DTCO].
- The protocol between the Application and the XA Resource Manager Bridge is implementation-specific.
- The interface between the XA Resource Manager Bridge and the XA Resource Manager is specified in [XOPEN-DTP].
- The process of enlisting a One-Pipe XA Resource Manager in an OleTx transaction involves creating an OleTx enlistment in the OleTx transaction. This enlistment makes the calls on the XA Resource Manager necessary for Two-Phase Commit according to the XA Protocol.

- After the XA Resource Manager is enlisted in the OleTx transaction, the application calls Create XID on the XA Resource Manager Bridge, passing in the OleTxtransaction and receiving a corresponding XID. This XID is then used to perform work on the XA Resource Manager.

1.3.2 Roles

This protocol specifies two additional roles, the XA Resource Manager Bridge role and the XA Superior Transaction Manager role, and extends the Transaction Manager role specified by [MS-DTCO]. These roles are described in the following sections.

1.3.2.1 XA Resource Manager Bridge Role

The XA Resource Manager Bridge role is typically responsible for performing the following tasks.

- Registration of a Two-Pipe XA Resource Manager
- Unregistration of a Two-Pipe XA Resource Manager
- Enlistment of a Two-Pipe XA Resource Manager on a transaction as a **Phase One** and **Phase Two** participant
- Registration of a One-Pipe XA Resource Manager
- Unregistration of a One-Pipe XA Resource Manager
- Enlistment of a One-Pipe XA Resource Manager on a transaction as a Phase One and Phase-Two participant.

1.3.2.2 XA Superior Transaction Manager Role

The XA Superior Transaction Manager role is typically responsible for performing the following tasks:

- Managing a mapping between XIDs provided by an XA Transaction Manager and the corresponding OleTx transactions.
- Participating in the XA Protocol coordinated by an XA Transaction Manager.
- Participating in a Two-Phase Commit coordinated by an XA Transaction Manager, and mapping to and from the related XA Protocol.
- Notifying a Transaction Manager of recovery initiated by an XA Transaction Manager, and participating in that process.

1.3.2.3 Transaction Manager Role

This document specifies the following **facet**, in addition to those specified in [MS-DTCO].

- XA Resource Manager Bridge Facet
- XA Subordinate Transaction Manager Facet

1.3.2.3.1 XA Resource Manager Bridge Facet

The XA Resource Manager Bridge Facet provides the following services to an XA Resource Manager Bridge:

- Registration of a Two-Pipe XA Resource Manager.

- Unregistration of a Two-Pipe XA Resource Manager.
- Enlistment of a Two-Pipe XA Resource Manager on a transaction as a Phase-One and Phase-Two participant.
- Registration of a One-Pipe XA Resource Manager.
- Unregistration of a One-Pipe XA Resource Manager.
- Participating in recovery and outcome notification for XA Resource Managers enlisted on a transaction.
- Participating in a Two-Phase Commit coordinated by a Transaction Manager, and mapping to and from the related XA Protocol

1.3.2.3.2 XA Subordinate Transaction Manager Facet

The XA Subordinate Transaction Manager Facet provides the following services to an XA Superior Transaction Manager.

- Creation of OleTx transactions mapped to XIDs.
- Participating in the Two-Phase Commit Protocol coordinated by an XA Transaction Manager, and mapping to and from the related XA Protocol.
- Participating in Recovery and outcome notification initiated by an XA Transaction Manager.

1.4 Relationship to Other Protocols

This protocol extends the protocol specified in [MS-DTCO]. The following table illustrates the protocol layering for this protocol:

- DTCXA
- DTCO
- CMP
- CMPO

1.5 Prerequisites/Preconditions

This protocol requires that all participating roles possess implementations of the MSDTC Connection Manager: OleTx Transports Protocol (as specified in [MS-CMPO]) and MSDTC Connection Manager: OleTx Multiplexing Protocol (as specified in [MS-CMP].) This protocol also requires that an implementation of the protocol specified in [MS-DTCO] is accessible using the protocols specified in [MS-CMPO] and [MS-CMP].

1.6 Applicability Statement

This protocol applies to scenarios where an XA Resource Manager and an implementation of the protocol specified by [MS-DTCO] are available.

This protocol applies to scenarios where an XA Transaction Manager and an implementation of the protocol specified by [MS-DTCO] are available.

This protocol requires network topologies where the [MS-CMPO] and [MS-CMP] protocols constitute a viable network transport for establishing many short-lived connection exchanges that accomplish specific tasks.

1.7 Versioning and Capability Negotiation

This section specifies the versioning and capability aspects of this protocol.

All of the versioning, versioning negotiation, and capability negotiation mechanisms specified in [MS-DTCO] section 1.7 are applicable to this protocol.

The value chosen for the protocol version MUST match the implementation's degree of support for specific connection and message types as specified in the following sections.

- Section 2.2.3.1 defines versioning details for connection types and message types specific to the XA Resource Manager Bridge Facet role.
- Section 2.2.3.1 defines versioning details for connection types and message types specific to the XA Resource Manager Bridge role.
- Section 2.2.4.1 defines versioning details for connection types and message types specific to the XA Subordinate Transaction Manager Facet role.
- Section 2.2.4.1 defines versioning details for connection types and message types specific to the XA Superior Transaction Manager role.

1.8 Vendor-Extensible Fields

This protocol has no vendor-extensible fields.

1.9 Standards Assignments

This protocol has no standards assignments.

2 Messages

The following sections specify the syntax of the messages that are transported, on the wire, by this protocol.

2.1 Transport

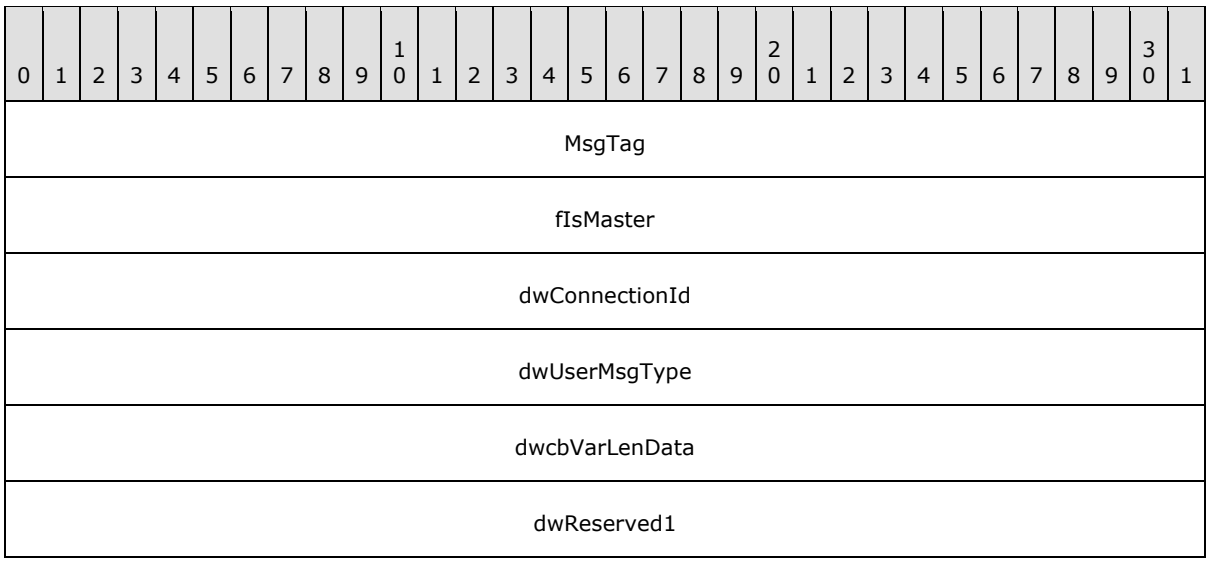
An implementation of this protocol uses the transport infrastructure provided by the underlying implementation of the MSDTC Connection Manager: OleTx Transaction Protocol, as specified in [MS-DTCO]. Therefore, the set of requirements specified in [MS-DTCO] section 2.1 MUST also apply to this protocol.

2.2 Message Syntax

2.2.1 Common Structures

2.2.1.1 MESSAGE_PACKET

The MESSAGE_PACKET structure defines the initial message fields that are contained by all **MTAGs** in this protocol, as specified in [MS-CMP] section 2.2.2.



MsgTag (4 bytes): A 4-byte integer value that describes the OLE transaction message type. For all uses in this document, this value MUST be MTAG_USER_MESSAGE, as specified in [MS-CMP section 2.2.8.

fIsMaster (4 bytes): A 4-byte value that indicates the direction of the message in the conversation.

This value MUST be one of the following values.

Value	Meaning
0x00000000	The message is sent by the party that accepted the connection.

Value	Meaning
0x00000001	The message is sent by the party that initiated the connection.

dwConnectionId (4 bytes): A 4-byte integer value that MUST contain the unique identifier for the associated connection.

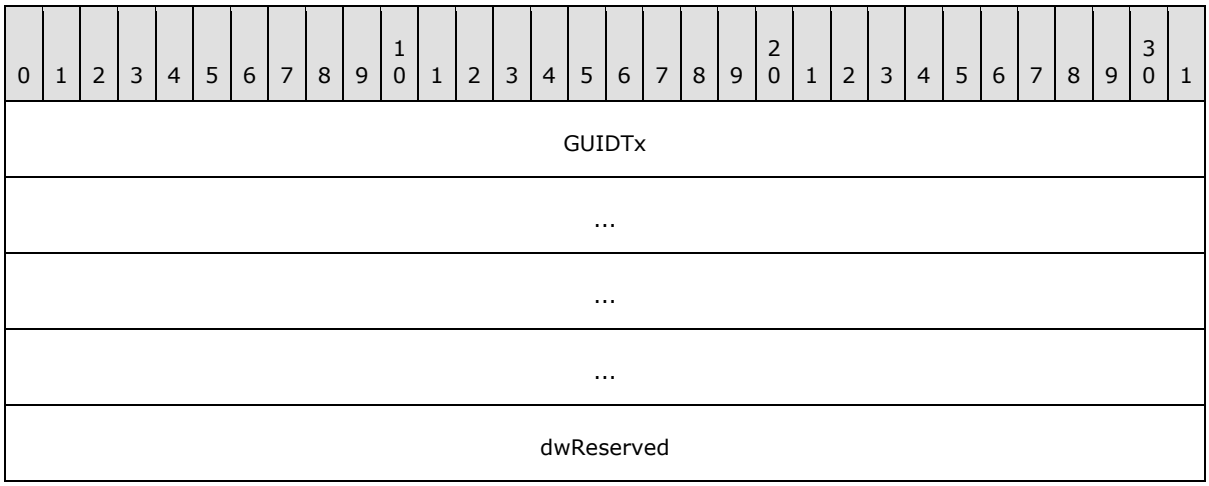
dwUserMsgType (4 bytes): This field contains the message type identifier. Each MTAG that is defined in this section MUST specify a distinct value for this field for a specified **connection type**

dwcbVarLenData (4 bytes): An unsigned 4-byte integer value that MUST contain the size, in bytes, of the message buffer that contains the MESSAGE_HEADER structure, minus the size, in bytes, of the MESSAGE_HEADER structure itself.

dwReserved1 (4 bytes): Reserved. This value MUST be set to an implementation-specific value and MUST be ignored on receipt.<1>

2.2.1.2 XA_GTRID

The XA_GTRID structure is used to represent the **XA Global Transaction Identifier** portion of an XA Transaction Branch Identifier.

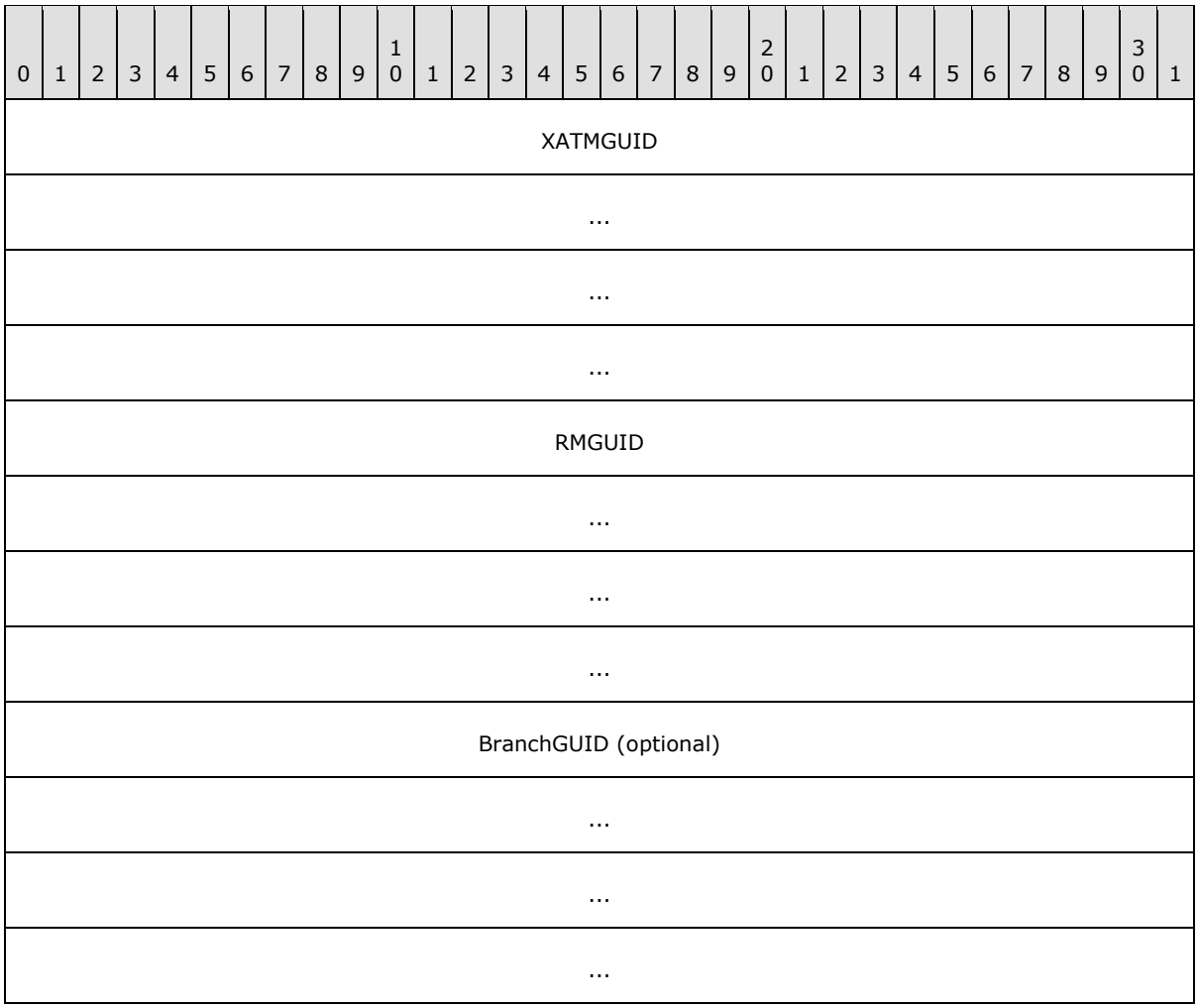


GUIDTx (16 bytes): This field MUST contain a **GUID** that specifies the transaction identifier.

dwReserved (4 bytes): Reserved. This value MUST be set to an implementation-specific value and MUST be ignored on receipt.

2.2.1.3 XA_BQUAL_1

The XA_BQUAL_1 structure is used to represent the **XA Branch Qualifier** portion of an XA Transaction Branch Identifier.



XATMGUID (16 bytes): This field MUST contain a **GUID** that specifies the **XA Transaction Manager Identifier** for the XA Transaction Manager instance.

RMGUID (16 bytes): This field MUST contain a **GUID** that specifies the Resource Manager Identifier for the XA Transaction Manager instance.

BranchGUID (16 bytes): If present, this field MUST contain a **GUID** that specifies the **XA Branch Identifier** for this **branch**.

2.2.1.4 XA_XID

The XA_XID structure is used to represent an **XA Transaction Branch**. This structure follows the format specified by the technical standard, "Distributed Transaction Processing: The XA Specification," as specified in [XOPEN-DTP].

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
formatID																															
gtridLength																															
bqualLength																															
Data																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(Data cont'd for 24 rows)																															

formatID (4 bytes): A 32-bit integer value that MUST contain an **XA Format Identifier** that identifies the format used to interpret the remainder of the XA_XID. It MUST be set to the following value.

0x00445443

gtridLength (4 bytes): An integer value that MUST contain the length in bytes of the XA Global Transaction Identifier portion of an XA Transaction Branch Identifier, as stored in the data array. This value MUST be no greater than 64 bytes.

Value	Meaning
16	The size in bytes of a **GUID**. This value is used if the value of **formatID** is 0x00445443.

bqualLength (4 bytes): An integer value that MUST contain the length, in bytes, of the XA Branch Qualifier portion of an XA Transaction Branch Identifier, as stored in the Data array. This value MUST be no greater than 64 bytes.

Value	Meaning
32	The size, in bytes, of a XA_BQUAL_1 structure. This value is used if the value of **formatID** is 0x00445443 and the **BranchGUID** field is not present in the XA_BQUAL_1 structure.
48	The size, in bytes, of a XA_BQUAL_1 structure. This value is used if the value of **formatID** is 0x00445443 and the **BranchGUID** field is present in the XA_BQUAL_1 structure.

Data (128 bytes): An array of bytes MUST contain both the XA Global Transaction Identifier and the XA Branch Qualifier portions of an XA Transaction Branch Identifier. The array MUST be 128 bytes in length. The XA Global Transaction Identifier data MUST begin at the first byte in the array. The XA Branch Qualifier data MUST begin at an offset of **gtridLength** bytes into the array. Bytes unused by the XA Global Transaction Identifier or the XA Branch Qualifier MUST be ignored.

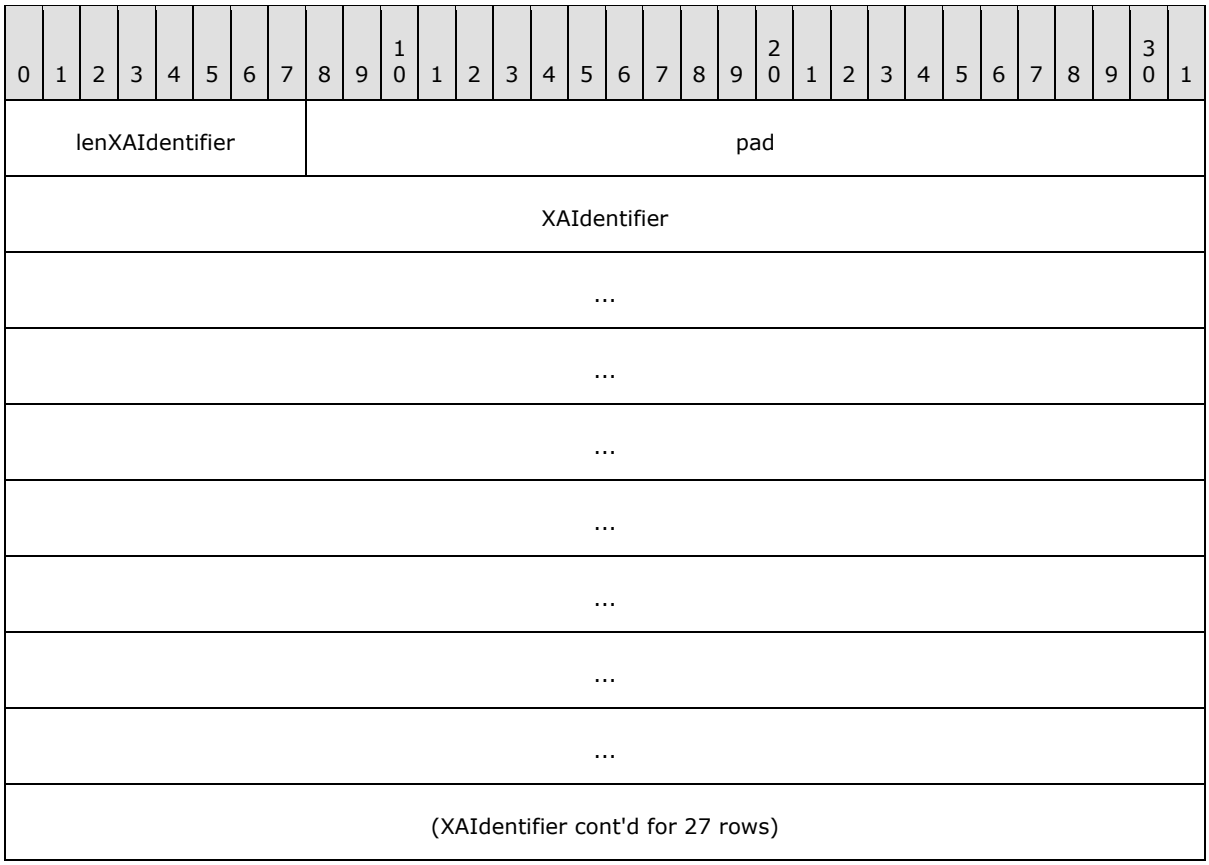
If the value of **formatID** is 0x00445443:

- The XA Global Transaction Identifier data MUST contain a **GUID** that specifies the transaction identifier.

The XA Branch Qualifier data MUST contain an XA_BQUAL_1 structure.

2.2.1.5 XA_UOW

The XA_UOW structure is used to represent a length-qualified XA Transaction Branch Identifier.



lenXAIdentifier (1 byte): An 8-bit unsigned value that MUST contain the length in bytes of the XAIdentifier field.

pad (3 bytes): A 3-byte padding field to align the following field to the next 4-byte boundary. Any padding bytes MUST be set to an implementation-specific value, and MUST be ignored on receipt.

XAIdentifier (140 bytes): This field MUST contain an XA_XID structure that specifies an XA Transaction Branch Identifier.

2.2.2 Enumeration

2.2.2.1 Connection Types

The CONNTYPE enumeration defines the connection types that are used by this protocol.

```
typedef  enum
{
  CONNTYPE_XATM_OPEN = 0x00001001,
  CONNTYPE_XATM_ENLIST = 0x00001002,
  CONNTYPE_XATM_OPENONEPIPE = 0x00001003,
  CONNTYPE_XAUSER_CONTROL = 0x00000040,
  CONNTYPE_XAUSER_XACT_START = 0x00000041,
  CONNTYPE_XAUSER_XACT_OPEN = 0x00000042,
  CONNTYPE_XAUSER_XACT_MIGRATE = 0x00000043,
```

```
  CONNTYPE_XAUSER_XACT_BRANCH_START = 0x00000050,
  CONNTYPE_XAUSER_XACT_BRANCH_OPEN = 0x00000051,
  CONNTYPE_XAUSER_XACT_MIGRATE2 = 0x00000052
} CONNTYPE;
```

2.2.3 Connection Types Relevant to XA Resource Manager Bridges and XA Resource Manager Bridge Facets

2.2.3.1 Versioning

The following table shows version-specific aspects for connection types that are relevant to XA Resource Manager Bridges and XA Resource Manager Bridge Facets. This table includes connection types and messages that are supported on certain versions as well as messages whose size is version-specific. If a connection type or message that is relevant to XA Resource Manager Bridge and XA Resource Manager Bridge Facet is omitted from this table, it is not version-specific and MUST be supported on all versions.

Version-specific aspect	Version 1	Version 2	Version 4	Version 5
Version supports MTAG XATMUSER_MTAG_E_CONFIGLOGWRITEFAILED.		X	X	X

2.2.3.2 CONNTYPE_XATM_OPEN

This connection type is used to register a Two-Pipe XA Resource Manager.

For more information about CONNTYPE_XATM_OPEN as an initiator and as an acceptor, see section Protocol Details.

2.2.3.2.1 XATMUSER_MTAG_E_CONFIGLOGWRITEFAILED

This message indicates that the request to register an XA Resource Manager failed because the Transaction Manager failed to durably record registration information.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000008.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.2.2 XATMUSER_MTAG_E_FAILEDRECOVERY

This message indicates that the request to register an XA Resource Manager failed because a previous attempt to recover has failed.

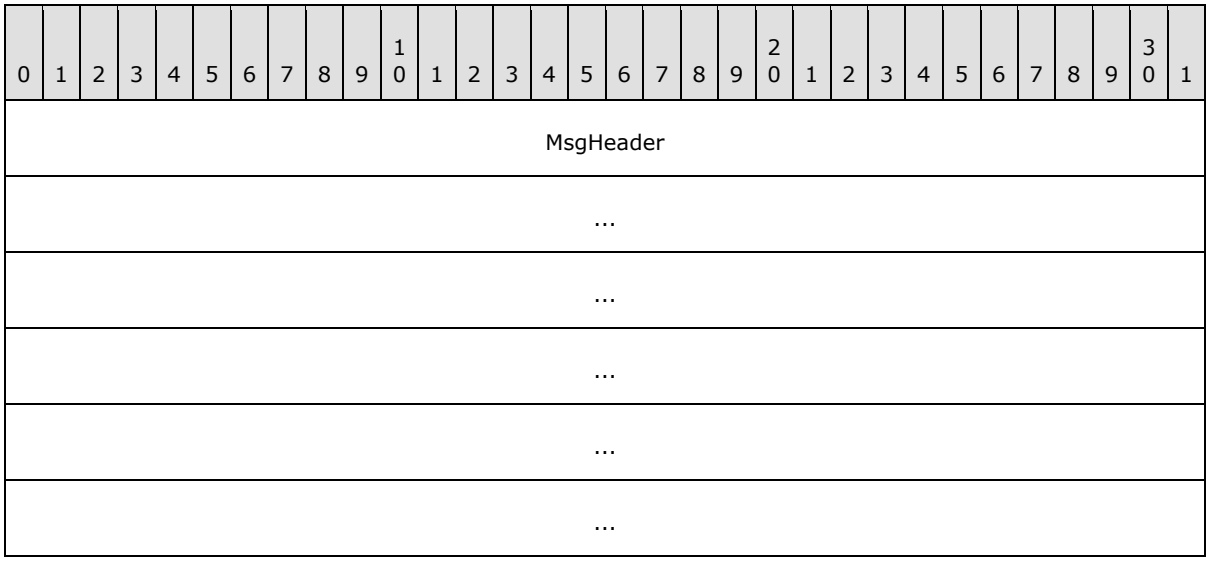
0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000006.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.2.3 XATMUSER_MTAG_E_RMNONEXISTENT

This message indicates that the request to register an XA Resource Manager failed because the XA Resource Manager was not found.

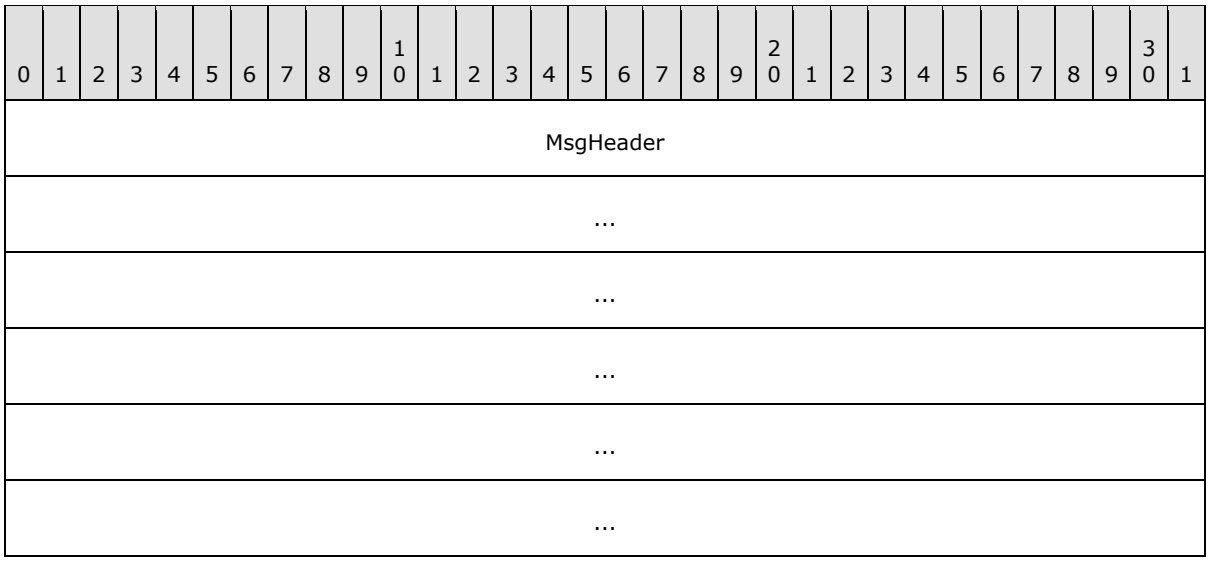


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000004.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.2.4 XATMUSER_MTAG_E_RMNOTAVAILABLE

This message indicates that the request to register an XA Resource Manager failed because the XA Resource Manager was unavailable.

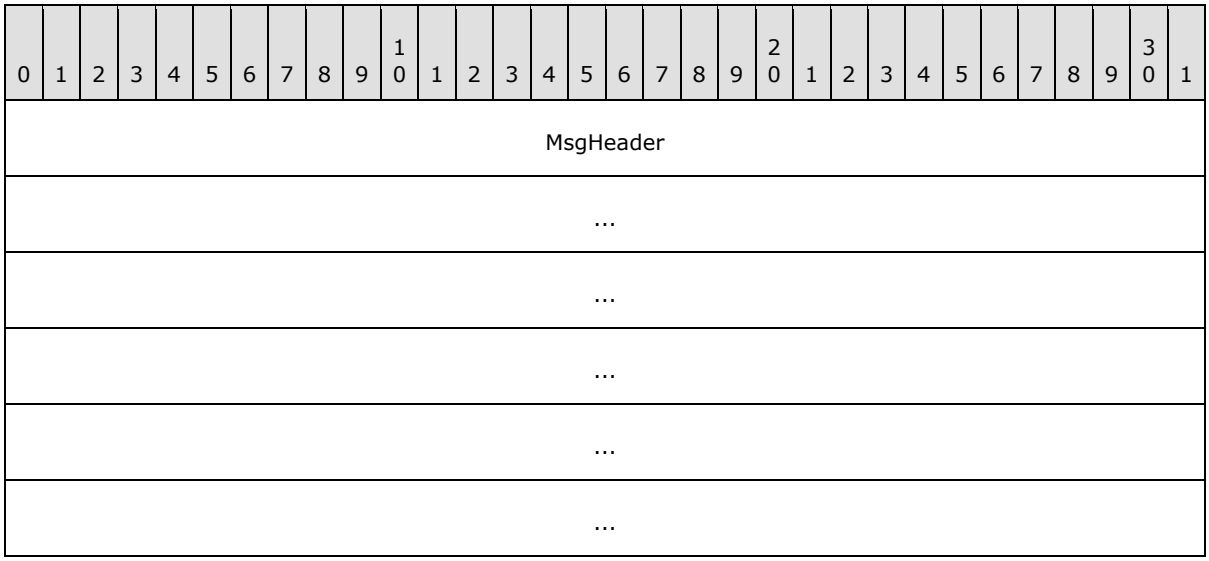


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000005.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.2.5 XATMUSER_MTAG_E_RMOPENFAILED

This message indicates that the request to register an XA Resource Manager failed for an unspecified reason.

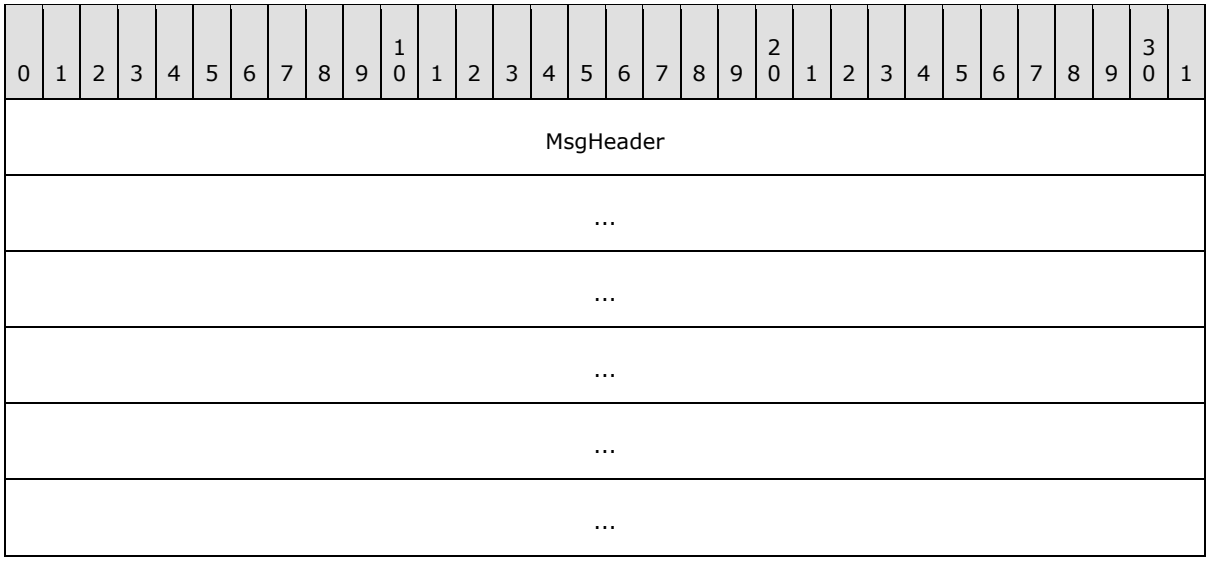


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000003.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.2.6 XATMUSER_MTAG_E_RMPROTOCOL

This message indicates that the request to register an XA Resource Manager failed because either the registration or a previous attempt to recover returned XAER_PROTO, as specified by the technical standard, "Distributed Transaction Processing: The XA Specification," as specified in [XOPEN-DTP].

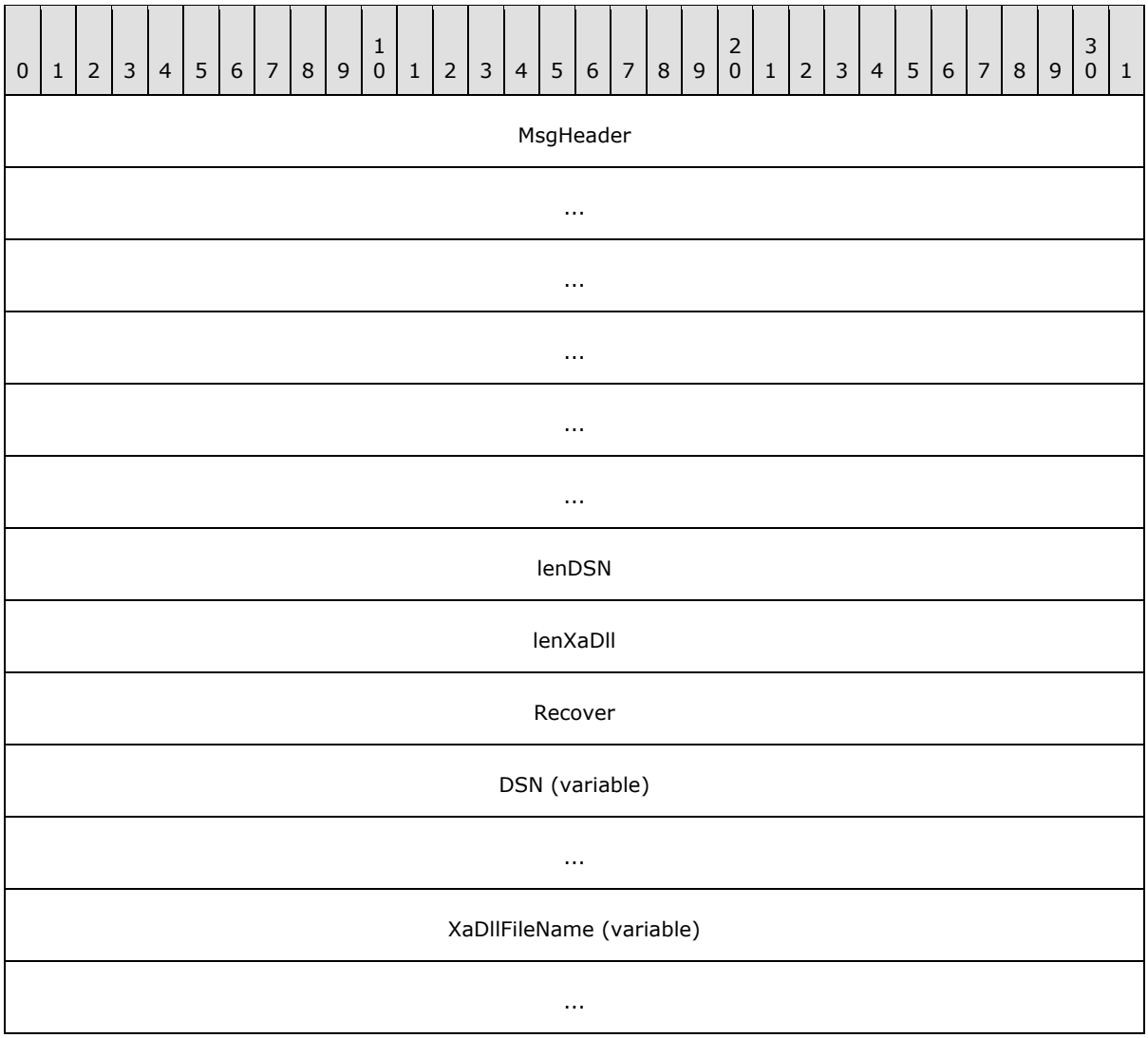


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000007.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.2.7 XATMUSER_MTAG_RMOPEN

This message requests the registration of an XA Resource Manager.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x20000001.
- The **dwcbVarLenData** field MUST be at least 12.

lenDSN (4 bytes): A 32-bit unsigned integer that MUST contain the count of bytes in the **DSN** string that follows this message. It MUST be less than 256.

lenXaDll (4 bytes): A 32-bit unsigned integer that MUST contain the count of bytes in the **XaDllFileName** string that follows this message. It MUST be less than 256.

Recover (4 bytes): This value MUST be one of the following values.

Value	Meaning
0x00000000	The transaction manager SHOULD NOT perform recovery.

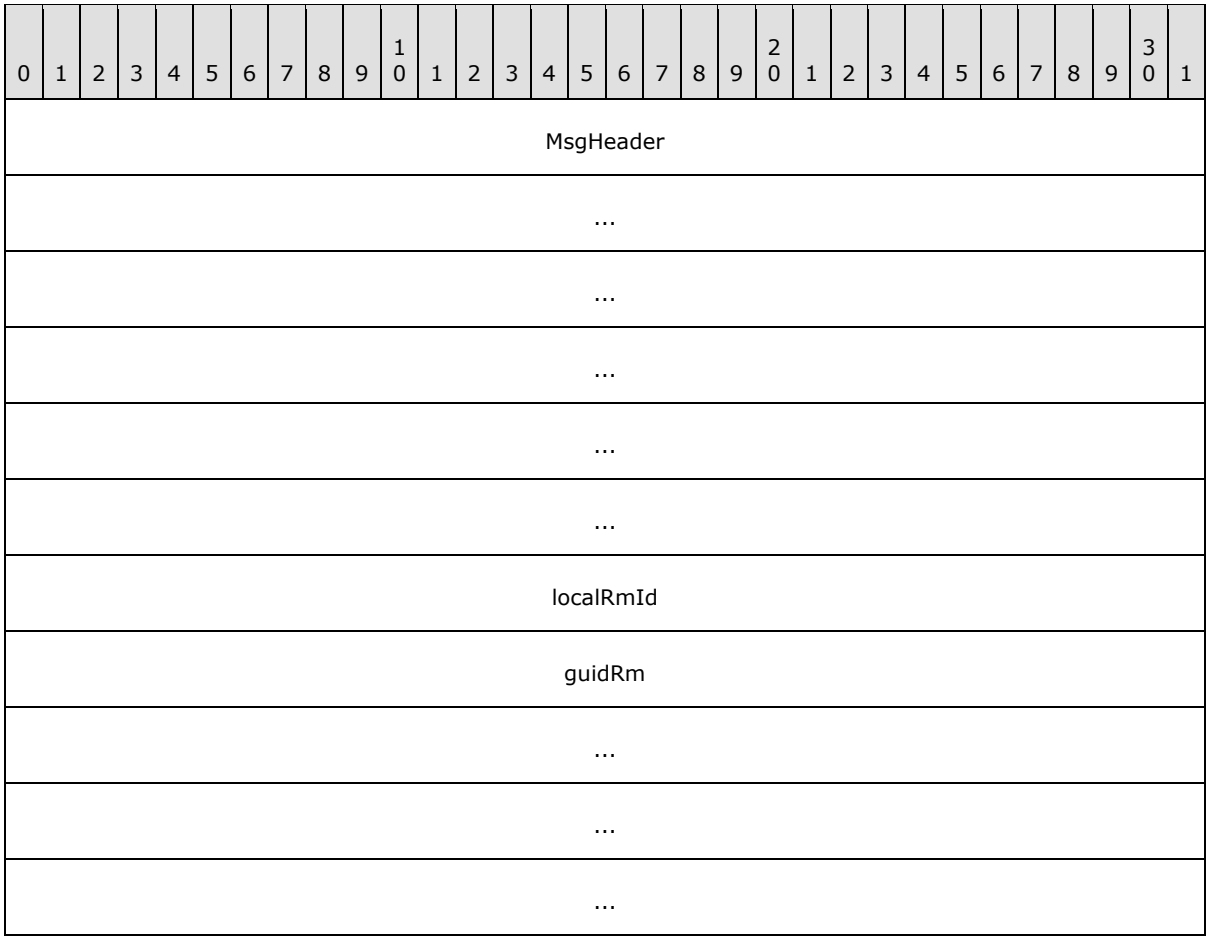
Value	Meaning
0x00000001	The transaction manager SHOULD perform recovery.

DSN (variable): A Latin-1 ANSI string that indicates the Data Source Name for the XA Resource Manager. It is not necessarily NULL-terminated. It MUST contain a number of bytes equal to the lenDSN field of the message.

XaDllFileName (variable): A Latin-1 ANSI string indicating the location of the XA DLL that SHOULD be loaded for the Resource Manager. It is not necessarily NULL-terminated. It MUST contain a number of bytes equal to the **lenXaDll** field of the message.

2.2.3.2.8 XATMUSER_MTAG_RMOPENOK

This message indicates the XA Resource Manager has been successfully registered.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x20000002.
- The **dwcbVarLenData** field MUST be 20.

localRmId (4 bytes): A 32-bit unsigned integer that MUST contain an **XA Resource Manager Instance Identifier**.

guidRm (16 bytes): This field MUST contain a **GUID** that specifies a Resource Manager Identifier for the XA Resource Manager.

2.2.3.3 CONNTYPE_XATM_OPENONEPIPE

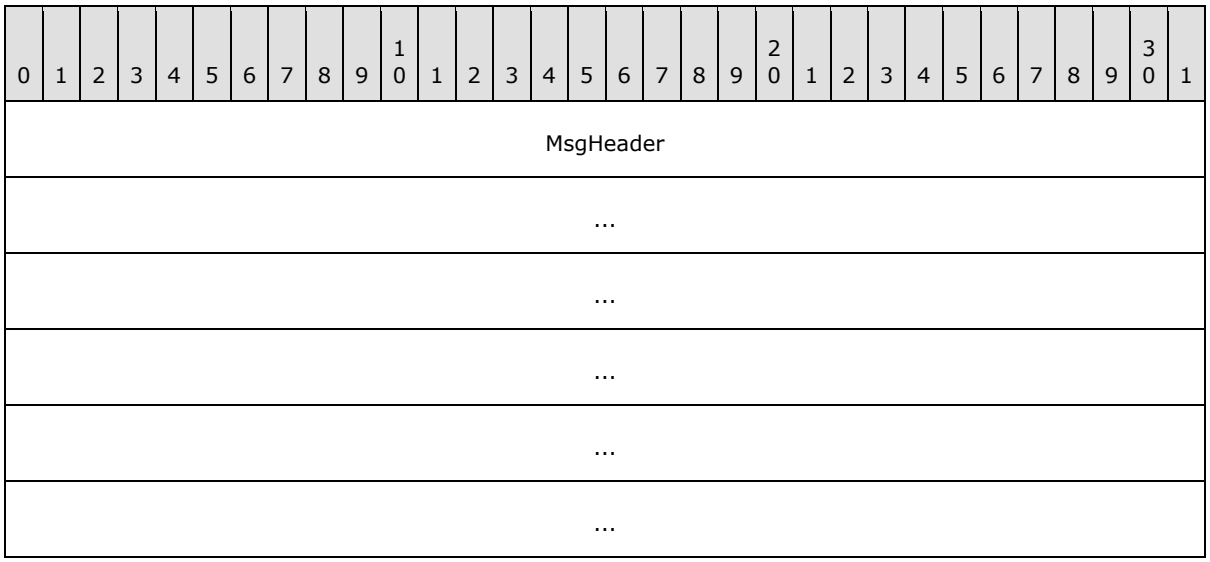
This connection type is used to register a one-pipe XA Resource Manager.

For more information about CONNTYPE_XATM_OPENONEPIPE as an initiator and as an acceptor, see section 3.

The same set of message types in section 2.2.2.1 are sent on this connection type.

2.2.3.3.1 XATMUSER_MTAG_E_CONFIGLOGWRITEFAILED

The XATMUSER_MTAG_E_CONFIGLOGWRITEFAILED packet indicates that the request to register an XA Resource Manager failed because the Transaction Manager failed to durably record registration information.

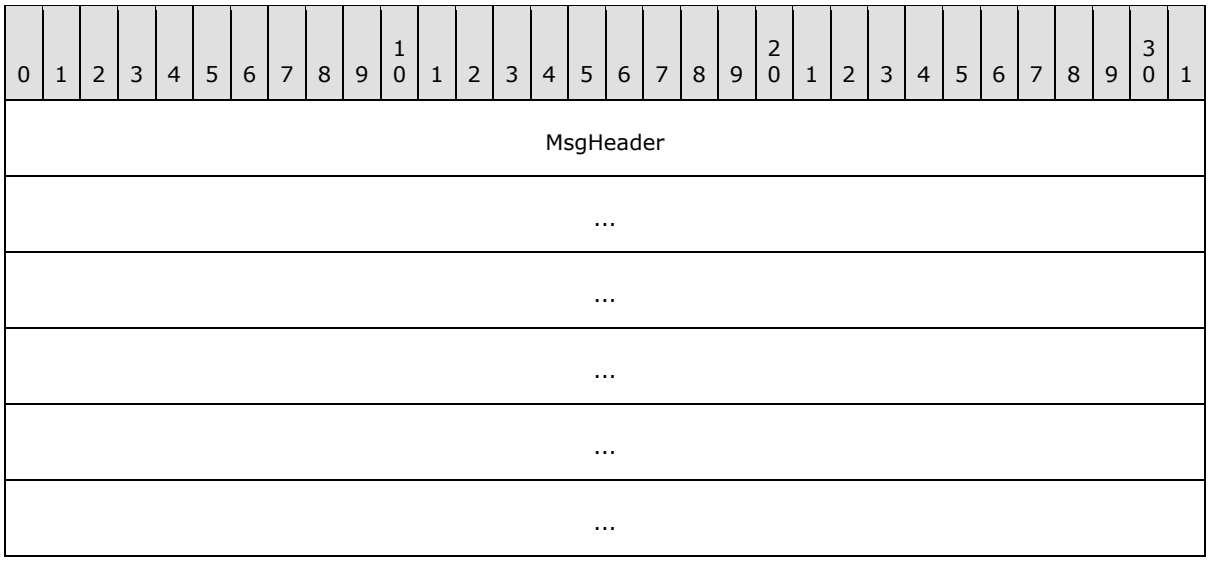


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000008.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.2 XATMUSER_MTAG_E_FAILEDRECOVERY

The XATMUSER_MTAG_E_FAILEDRECOVERY packet message indicates that the request to register an XA Resource Manager failed because a previous attempt to recover has failed.

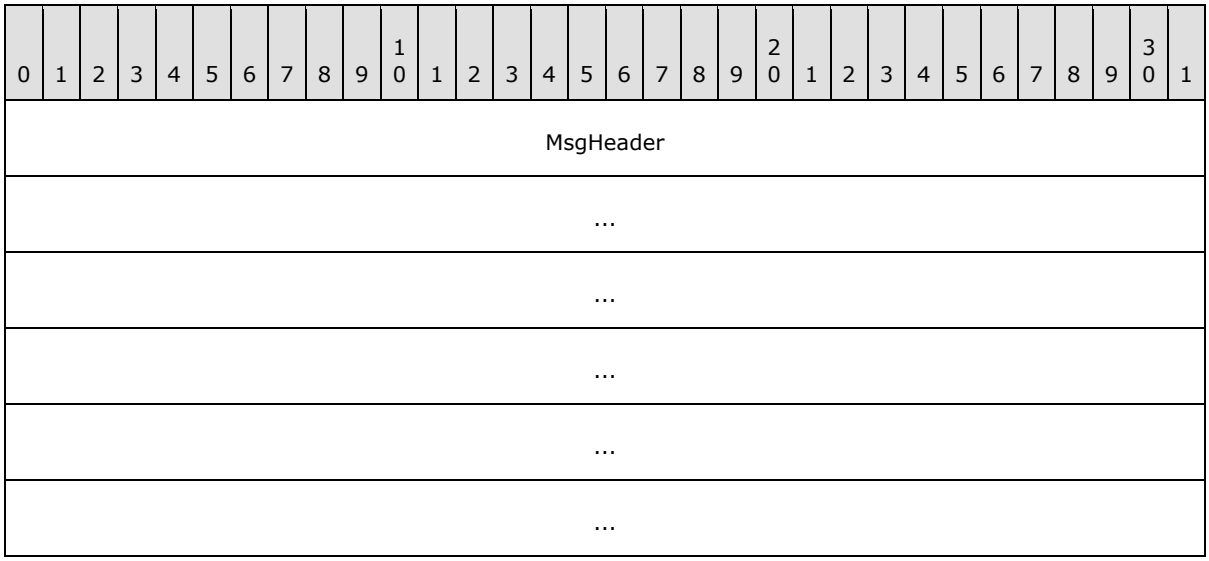


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000006.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.3 XATMUSER_MTAG_E_RMNONEXISTENT

The XATMUSER_MTAG_E_RMNONEXISTENT packet indicates that the request to register an XA Resource Manager failed because the XA Resource Manager was not found.

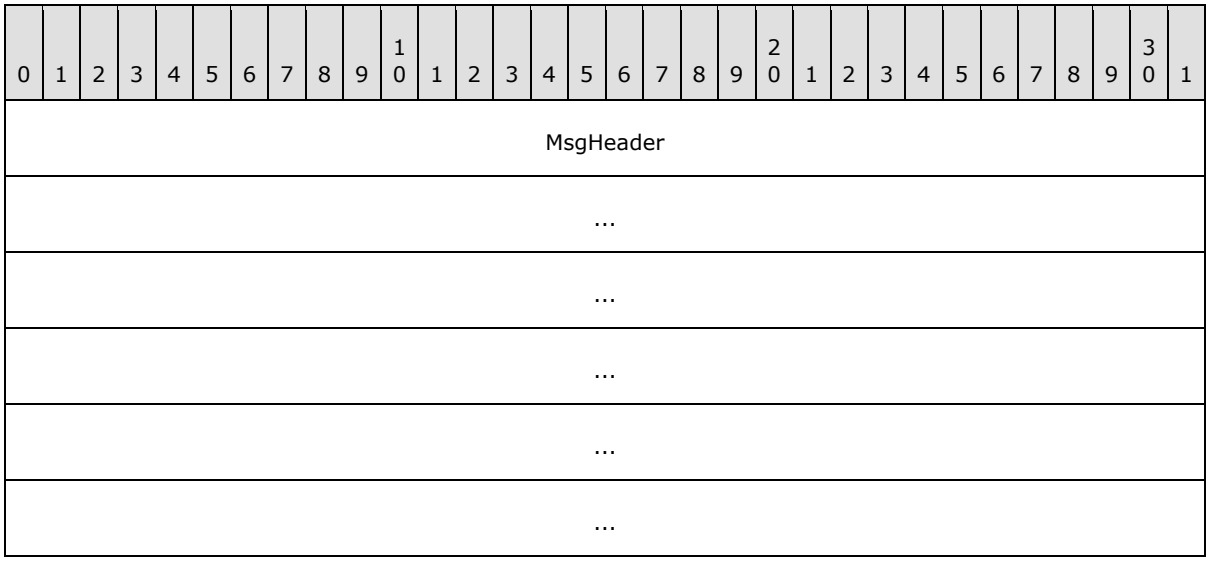


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000004.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.4 XATMUSER_MTAG_E_RMNOTAVAILABLE

The XATMUSER_MTAG_E_RMNOTAVAILABLE packet indicates that the request to register an XA Resource Manager failed because the XA Resource Manager was unavailable.

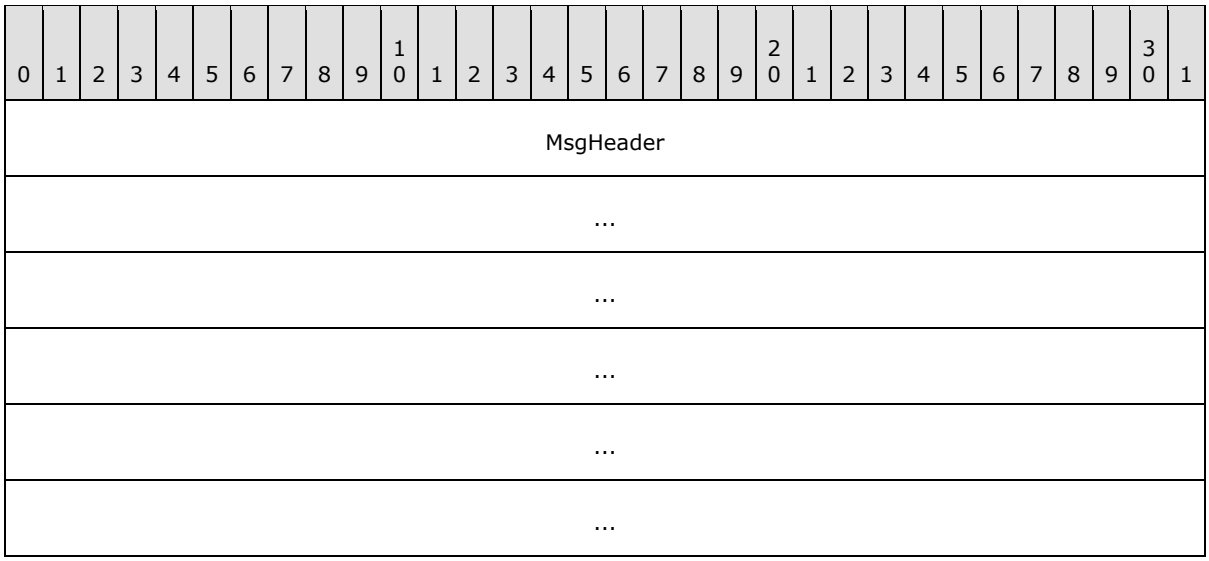


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000005.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.5 XATMUSER_MTAG_E_RMOPENFAILED

The XATMUSER_MTAG_E_RMOPENFAILED packet indicates that the request to register an XA Resource Manager failed for an unspecified reason.

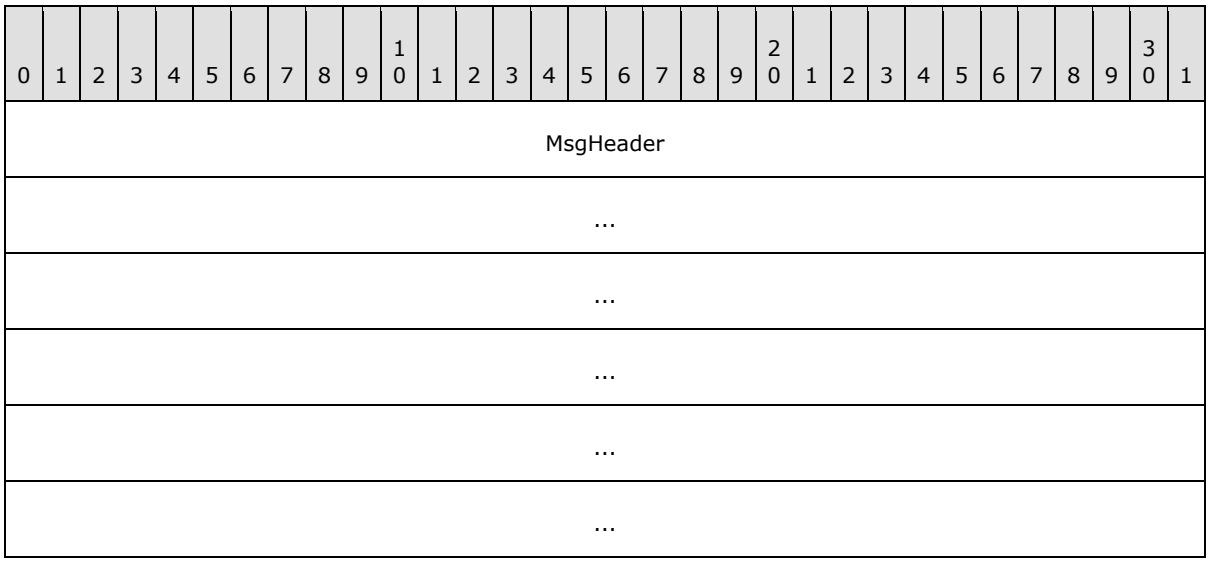


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000003.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.6 XATMUSER_MTAG_E_RMPROTOCOL

The XATMUSER_MTAG_E_RMPROTOCOL packet indicates that the request to register an XA Resource Manager failed because either the registration or a previous attempt to recover returned XAER_PROTO, as specified by the technical standard, "Distributed Transaction Processing: The XA Specification," as specified in [XOPEN-DTP].



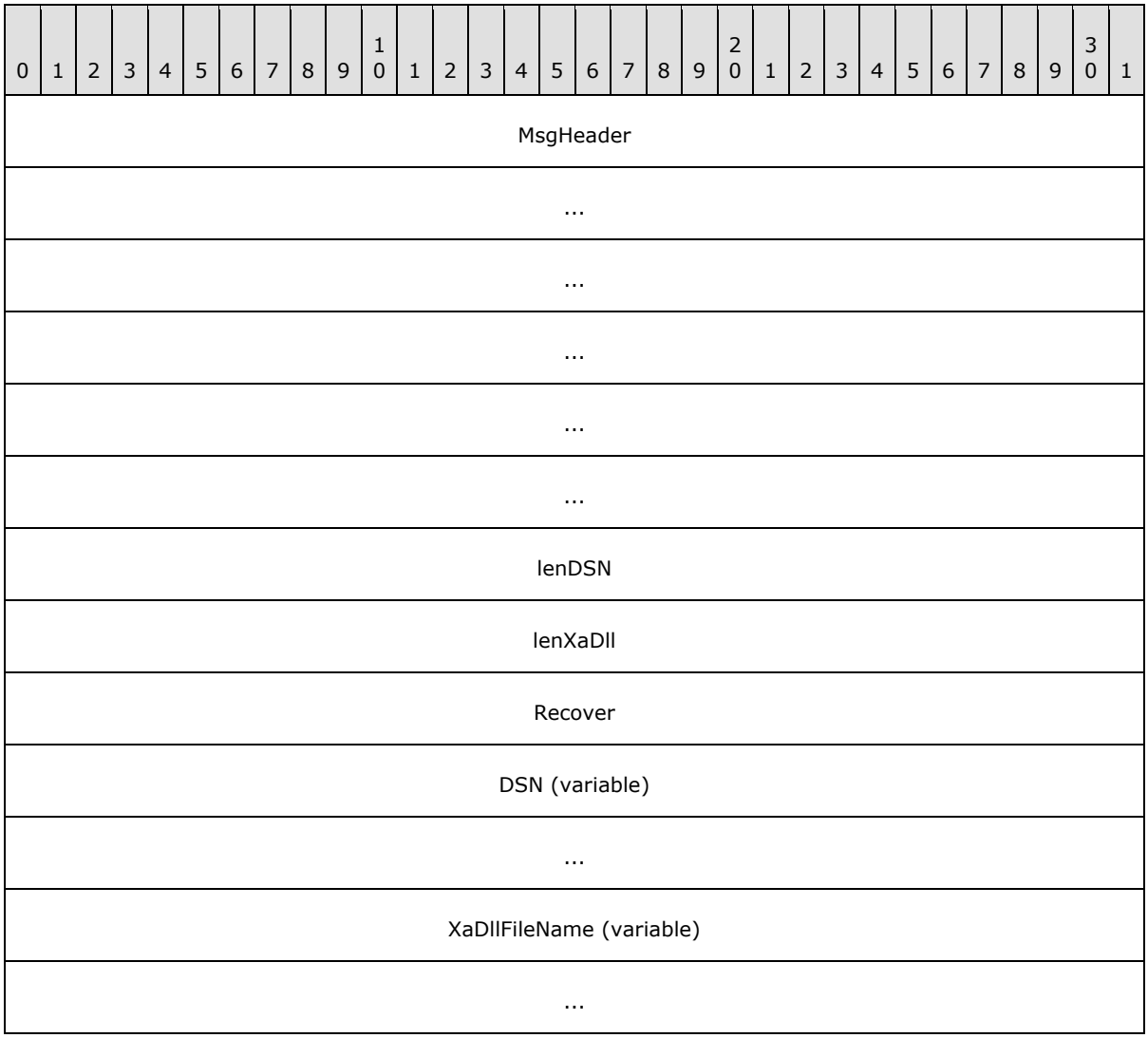
MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xA0000007.

- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.7 XATMUSER_MTAG_RMOPEN

The XATMUSER_MTAG_RMOPEN packet requests the registration of an XA Resource Manager.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x20000001.
- The **dwcbVarLenData** field MUST be at least 12.

lenDSN (4 bytes): A 32-bit unsigned integer that MUST contain the count of bytes in the **DSN** string that follows this message. It MUST be greater than 0 and less than 256.

lenXaDll (4 bytes): A 32-bit unsigned integer that MUST contain the count of bytes in the **XaDllFileName** string that follows this message. It MUST be less than 256.

Recover (4 bytes): A 4-byte value indicating whether the Transaction Manager SHOULD perform recovery when opening the Resource Manager.

This value MUST be one of the following values.

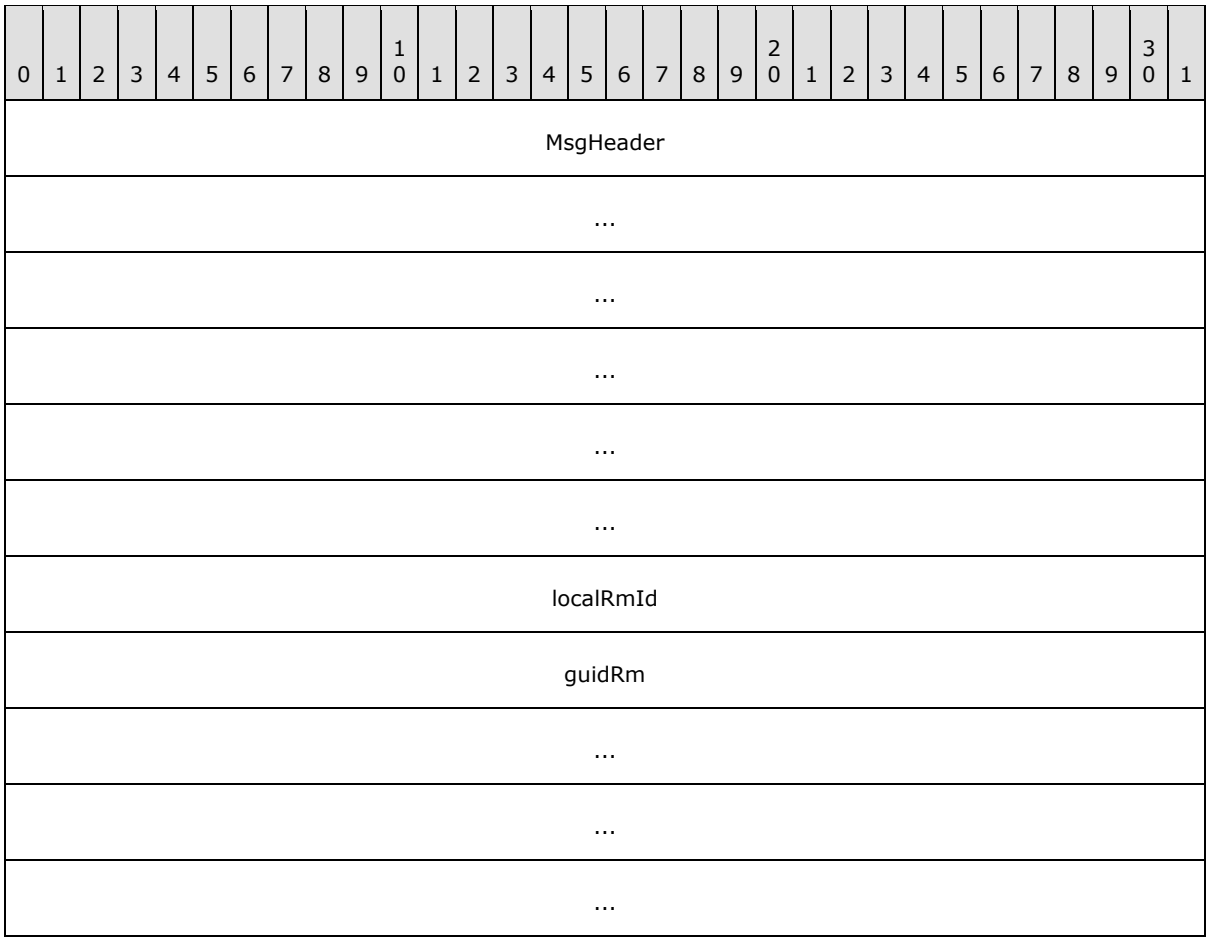
Value	Meaning
0x00000000	The Transaction Manager SHOULD NOT perform recovery.
0x00000001	The Transaction Manager SHOULD perform recovery.

DSN (variable): A Latin-1 ANSI string that indicates the Data Source Name for the XA Resource Manager. It is not necessarily NULL-terminated. It MUST contain a number of bytes equal to the **lenDSN** field of the message.

XaDllFileName (variable): A Latin-1 ANSI string indicating the location of the XA DLL that SHOULD be loaded for the resource manager. It is not necessarily NULL-terminated. It MUST contain a number of bytes equal to the **lenXaDll** field of the message.

2.2.3.3.8 XATMUSER_MTAG_RMOPENOK

The XATMUSER_MTAG_RMOPENOK packet indicates the XA Resource Manager has been successfully registered.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x20000002.
- The **dwcbVarLenData** field MUST be 20.

localRmId (4 bytes): A 32-bit unsigned integer that MUST contain an XA Resource Manager Instance Identifier.

guidRm (16 bytes): This field MUST contain a **GUID** that specifies a Resource Manager Identifier for the XA Resource Manager.

2.2.3.3.9 XATMUSER_MTAG_E_RMCLOSEFAILED

This message indicates that the request to unregister an XA Resource Manager failed because of an unspecified reason.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x90000003.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.10 XATMUSER_MTAG_E_RMCLOSEPROTOCOL

This message indicates that the request to unregister an XA Resource Manager failed because the XA Resource Manager returned XAER_PROTO, as specified by the technical standard, "Distributed Transaction Processing: The XA Specification," as specified in [XOPEN-DTP].

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

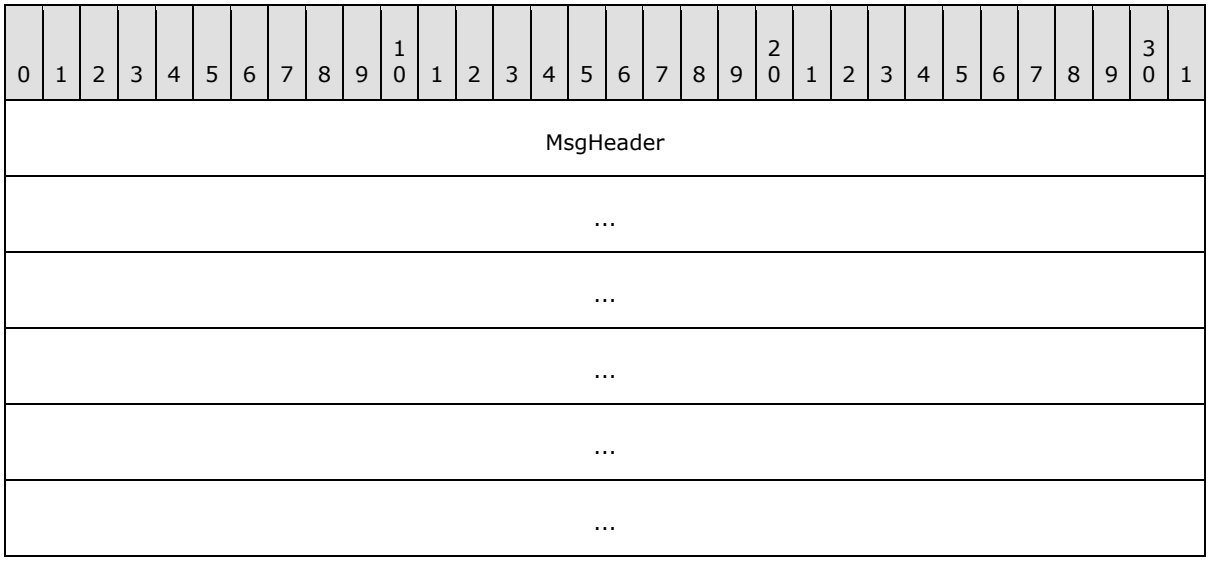
MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x90000008.

- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.11 XATMUSER_MTAG_E_RMCLOSERMNOTAVAILABLE

This message indicates that the request to unregister an XA Resource Manager failed because the XA Resource Manager was no longer available.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x90000004.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.12 XATMUSER_MTAG_E_RMCLOSETMERROR

This message indicates that the request to unregister an XA Resource Manager failed because the Transaction Manager reported an error.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x90000006.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.13 XATMUSER_MTAG_E_RMCLOSETMNOTAVAILABLE

This message indicates that the request to unregister an XA Resource Manager failed because the Transaction Manager was no longer available.

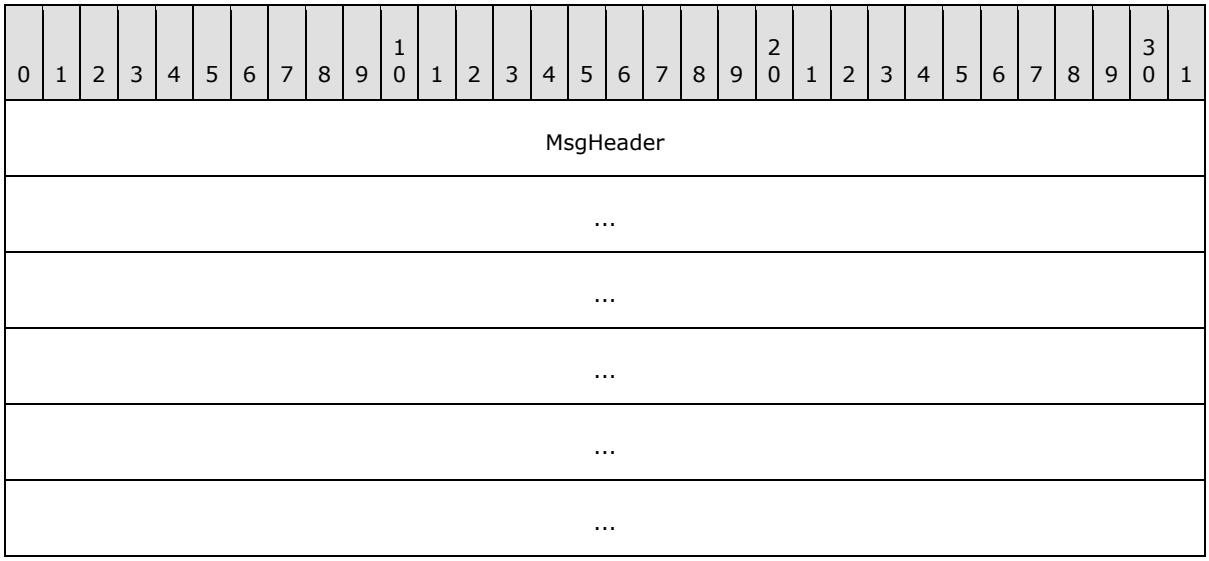
0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x90000005.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.14 XATMUSER_MTAG_E_RMCLOSEUNEXPECTED

This message indicates that the request to unregister an XA Resource Manager failed due to an unexpected error.

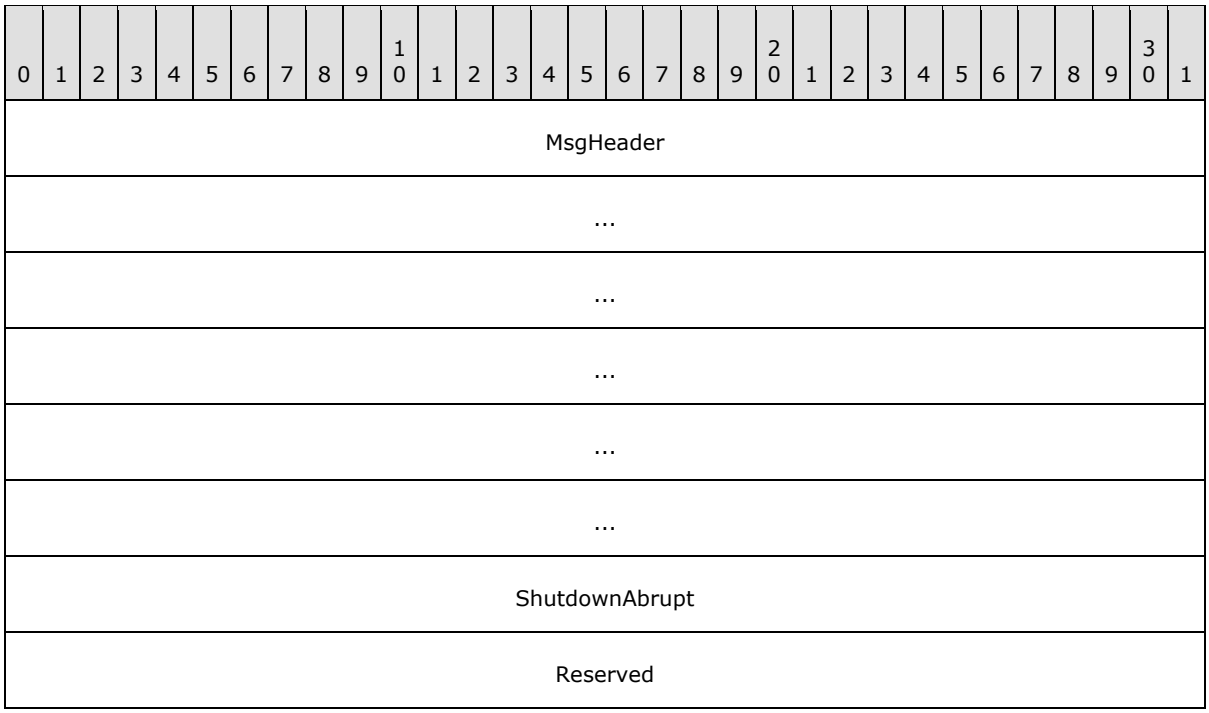


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x90000007.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.3.15 XATMUSER_MTAG_RMCLOSE

This message requests to unregister an XA Resource Manager.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x10000001.
- The **dwcbVarLenData** field MUST be 8.

ShutdownAbrupt (4 bytes): A 32-bit unsigned integer that MUST contain one of the following values to indicate whether this close represents an abrupt termination of the XA Resource Manager.

Value	Meaning
0x00000000	The shutdown is not abrupt.
0x00000001	The shutdown is abrupt.

Reserved (4 bytes): Reserved. This value MUST be set to an implementation-specific value and MUST be ignored on receipt.

2.2.3.3.16 XATMUSER_MTAG_RMCLOSEOK

This message indicates that the XA Resource Manager was unregistered successfully.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x10000002.
- The **dwcbVarLenData** field MUST be 0.

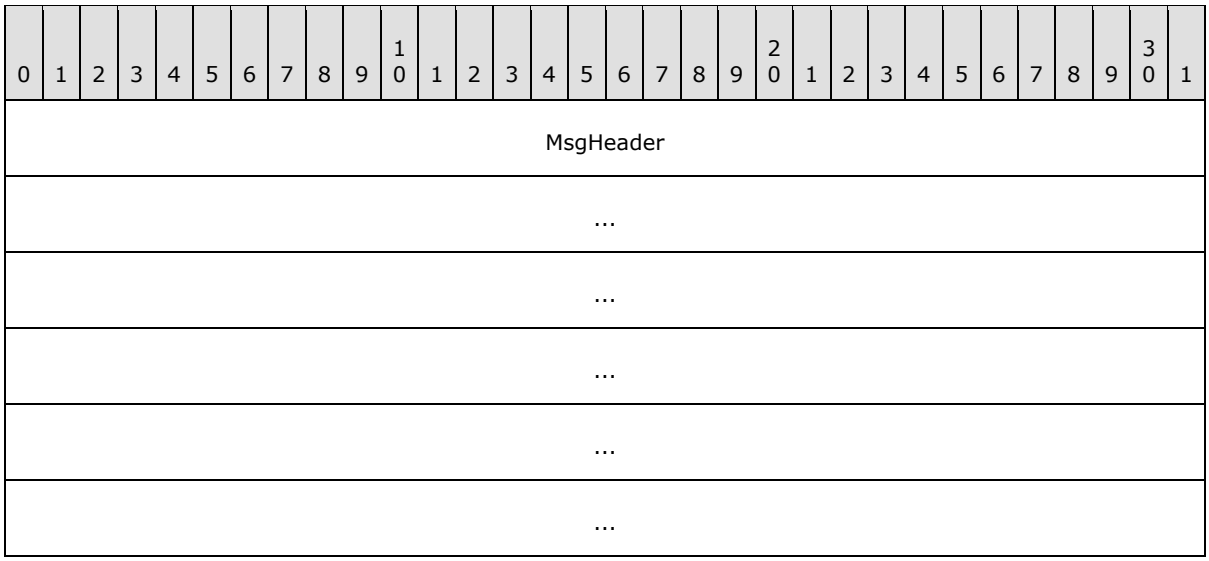
2.2.3.4 CONNTYPE_XATM_ENLIST

This connection type is used to enlist an XA Resource Manager into a transaction.

For more information about CONNTYPE_XATM_ENLIST as an initiator and as an acceptor, see section 3.

2.2.3.4.1 XATMUSER_MTAG_E_ENLISTMENTDUPLICATE

This message indicates that the enlistment has failed because the specified XA Resource Manager is already enlisted on the transaction.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xC0000006.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.4.2 XATMUSER_MTAG_E_ENLISTMENTFAILED

This message indicates that the enlistment has failed for an unspecified reason.

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xC0000005.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.4.3 XATMUSER_MTAG_E_ENLISTMENTIMPFAILED

This message indicates that the enlistment has failed because the Transaction Manager was unable to import the transaction specified by the **ImportCookie** member of the XATMUSER_MTAG_ENLIST message.

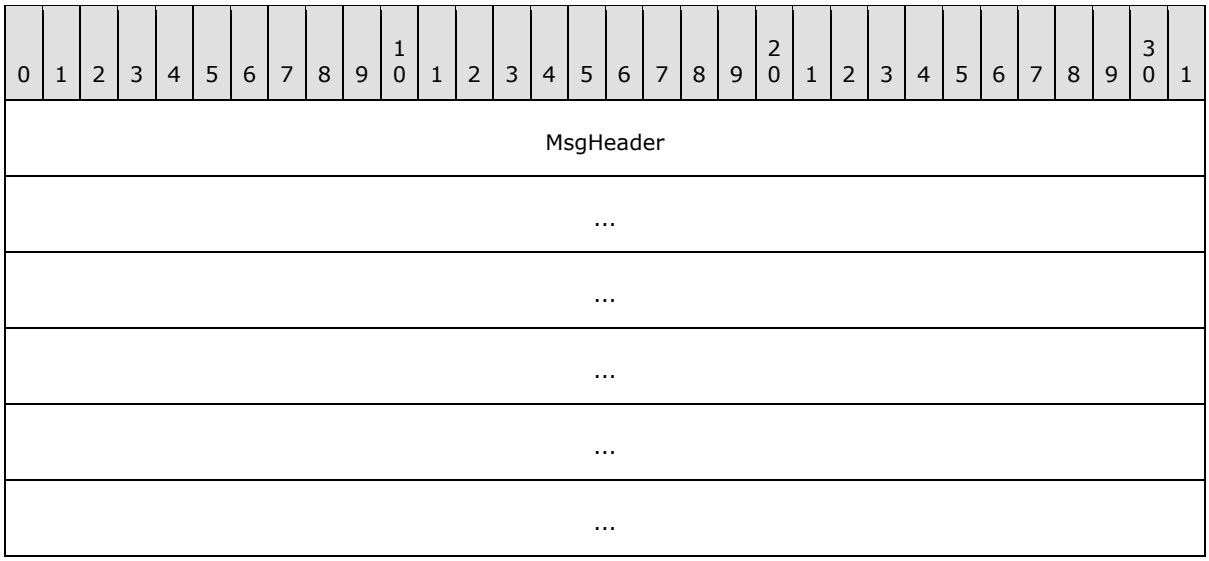
0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xC0000004.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.4.4 XATMUSER_MTAG_E_ENLISTMENTNOMEMORY

This message indicates that the enlistment has failed because the Transaction Manager was unable to allocate sufficient memory to perform the requested operation.

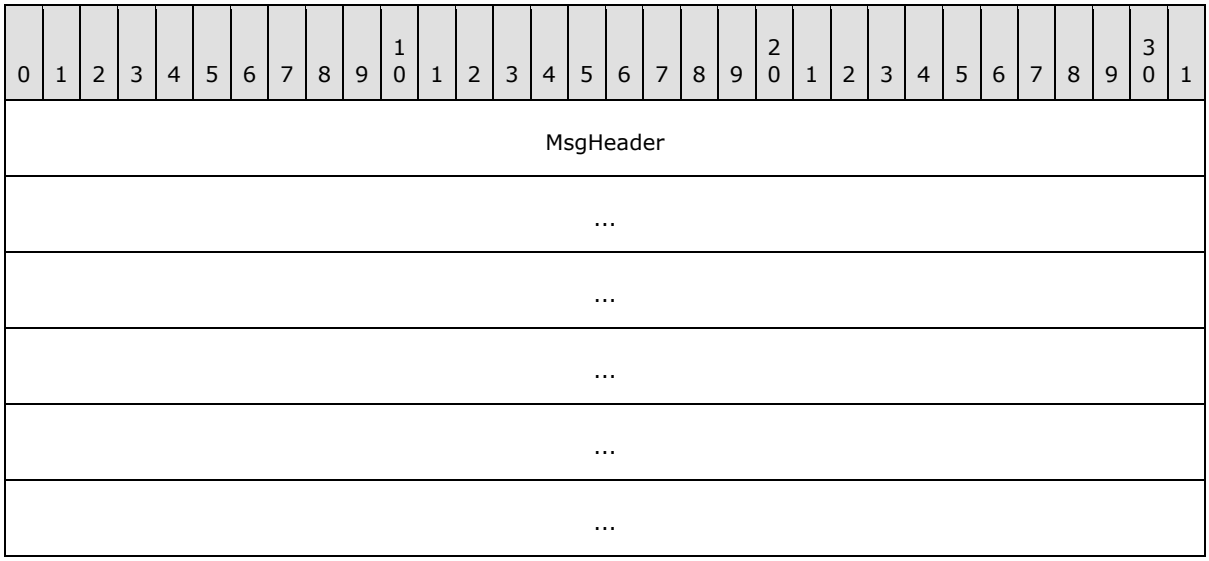


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xC0000007.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.4.5 XATMUSER_MTAG_E_ENLISTMENTRMNOTFOUND

This message indicates that the enlistment has failed because the enlisting XA Resource Manager was not found.

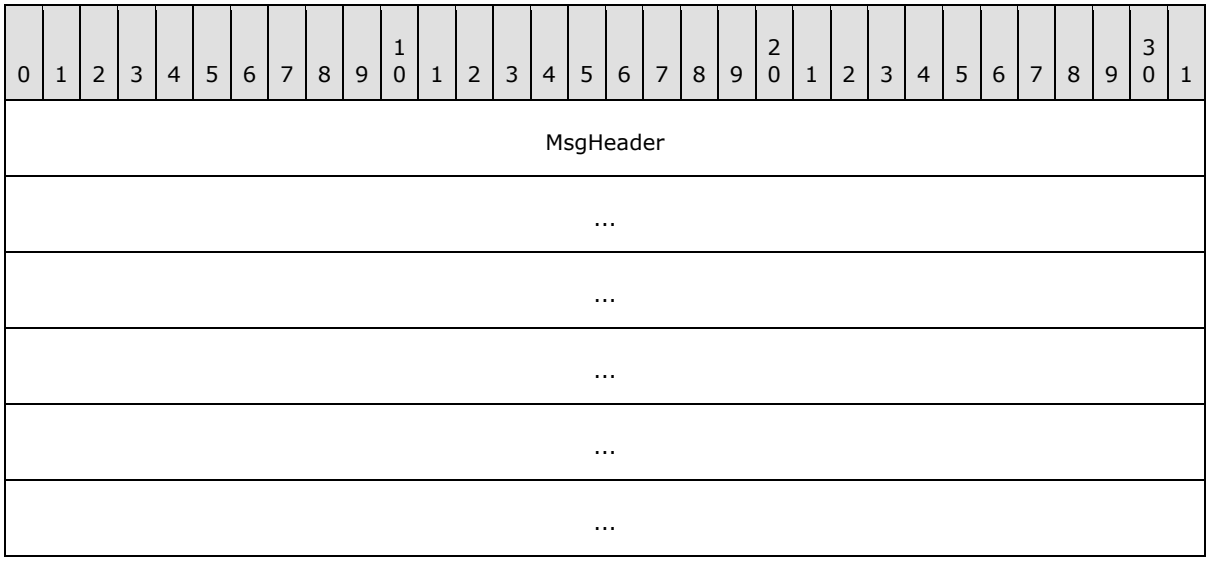


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xC0000003.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.4.6 XATMUSER_MTAG_E_ENLISTMENTRMRECOVERING

This message indicates that the enlistment has failed because the Transaction Manager is currently performing transaction recovery with the enlisting XA Resource Manager.

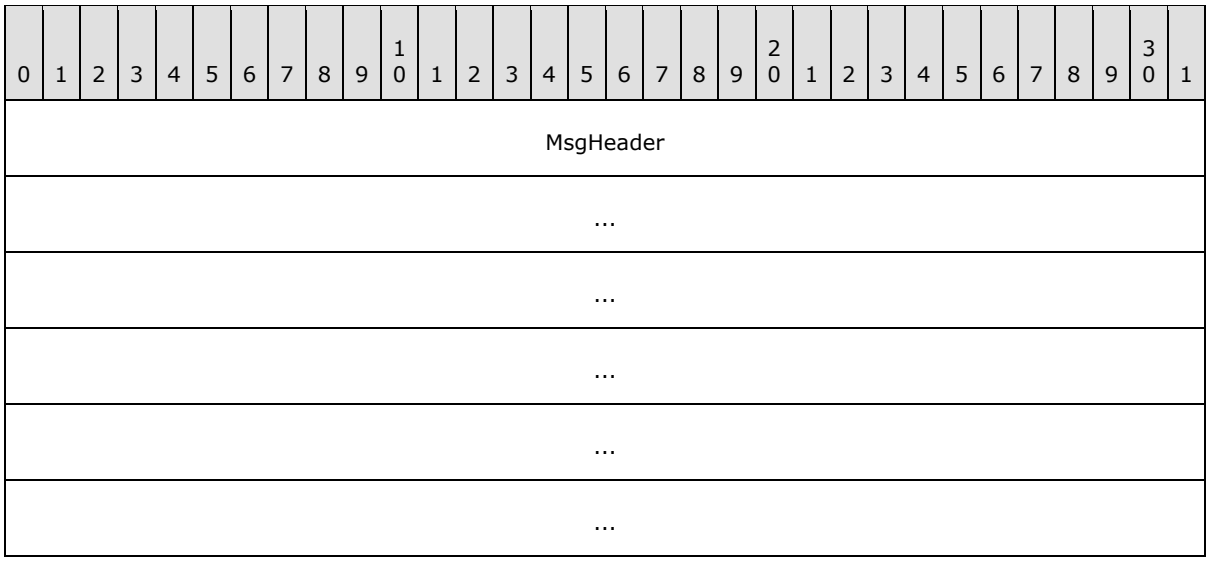


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xC0000009.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.4.7 XATMUSER_MTAG_E_ENLISTMENTRMUNAVAILABLE

This message indicates that the enlistment has failed because the enlisting XA Resource Manager is no longer available to complete the enlistment.

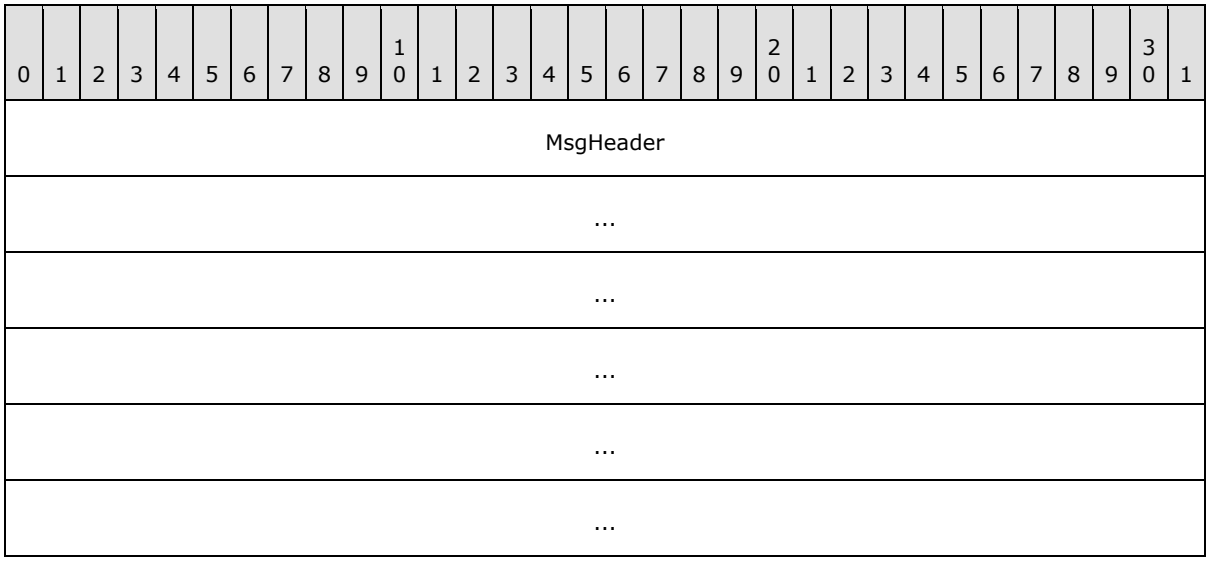


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xC000000A.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.4.8 XATMUSER_MTAG_E_ENLISTMENTTOOLATE

This message indicates that the enlistment has failed because the transaction is no longer in the Active or Phase 0 state, and therefore no longer able to accept new enlistments.

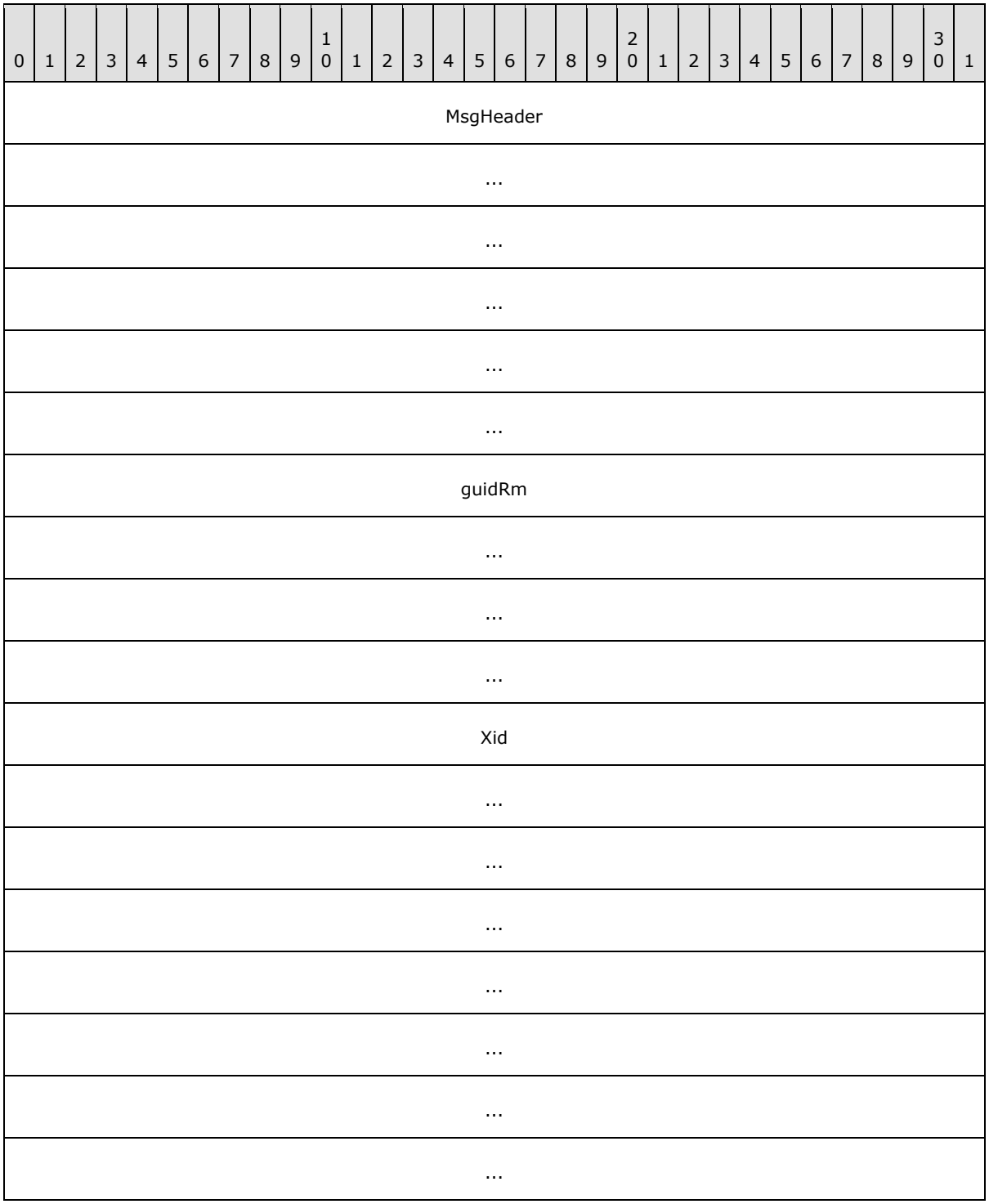


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0xC0000008.
- The **dwcbVarLenData** field MUST be 0.

2.2.3.4.9 XATMUSER_MTAG_ENLIST

This message enlists an XA Resource Manager into a transaction.



(Xid cont'd for 27 rows)
lenImportCookie
ImportCookie (variable)
...

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x40000001.
- The **dwcbVarLenData** field MUST be greater than 160 (160 bytes, plus the amount for the import cookie member, following the **cbImportCookie** field).

guidRm (16 bytes): This field MUST contain a **GUID** that specifies a resource manager identifier of the XA Resource Manager applying to enlist. There MUST be an open XA Resource Manager that is identified by this resource manager identifier.

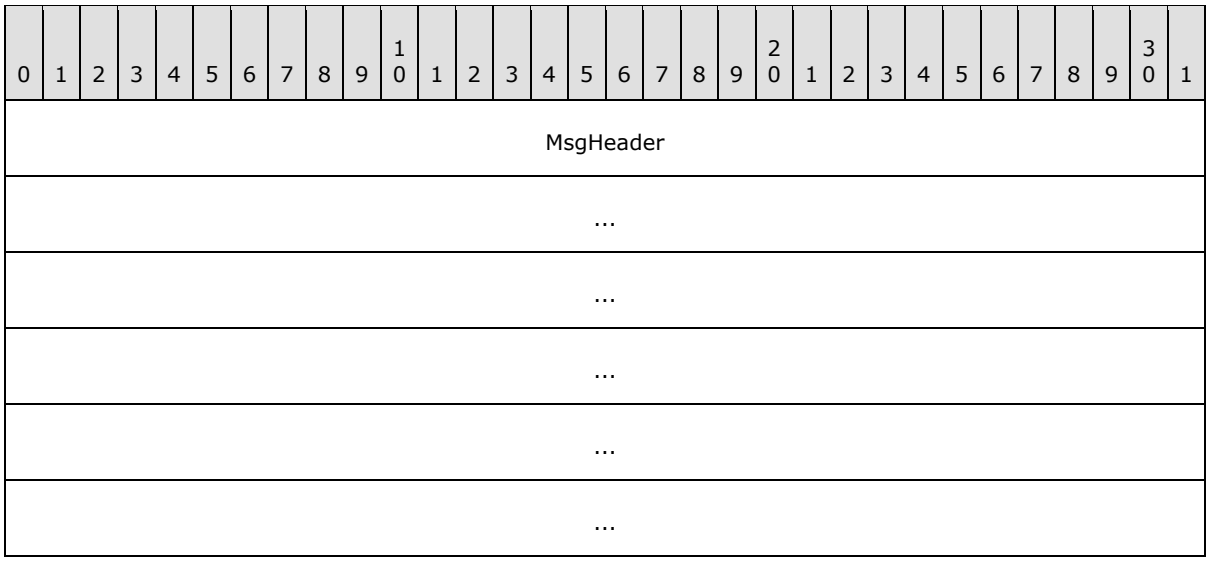
Xid (140 bytes): This field MUST contain an XA_XID structure that specifies an XA Transaction Branch Identifier.

lenImportCookie (4 bytes): A 32-bit unsigned integer that MUST contain the size, in bytes, of the import cookie, which is used to determine the transaction in which to enlist.

ImportCookie (variable): This field MUST be a STxInfo structure that specifies the transaction to enlist in. It is present only if the value of the **lenImportCookie** is nonzero.

2.2.3.4.10 XATMUSER_MTAG_ENLISTMENTOK

This message informs the XA Resource Manager that the enlistment has been successfully processed.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x40000002.
- The **dwcbVarLenData** field MUST be 0.

2.2.4 Connection Types Relevant to XA Superior Transaction Managers and XA Subordinate Transaction Manager Facets

2.2.4.1 Versioning

The following table shows version-specific aspects for connection types that are relevant to XA Superior Transaction Managers and XA Subordinate Transaction Manager Facets. This table includes connection types and messages that are supported on certain versions as well as messages whose size is version-specific. If a connection type or message that is relevant to XA Superior Transaction Managers and XA Subordinate Transaction Manager Facets is omitted from this table, it is not version-specific and MUST be supported on all versions.

Version-specific aspect	Version 1	Version 2	Version 4	Version 5
Version supports connection type. CONNTYPE_XA_USER_XACT_MIGRATE.		X<2>	X	X
Version supports connection type. CONNTYPE_XA_USER_XACT_MIGRATE2.				X<3>
Version supports connection type. CONNTYPE_XA_USER_XACT_BRANCH_START.				X<4>
Version supports connection type. CONNTYPE_XA_USER_XACT_BRANCH_OPEN.				X<5>

2.2.4.2 CONNTYPE_XAUSER_CONTROL

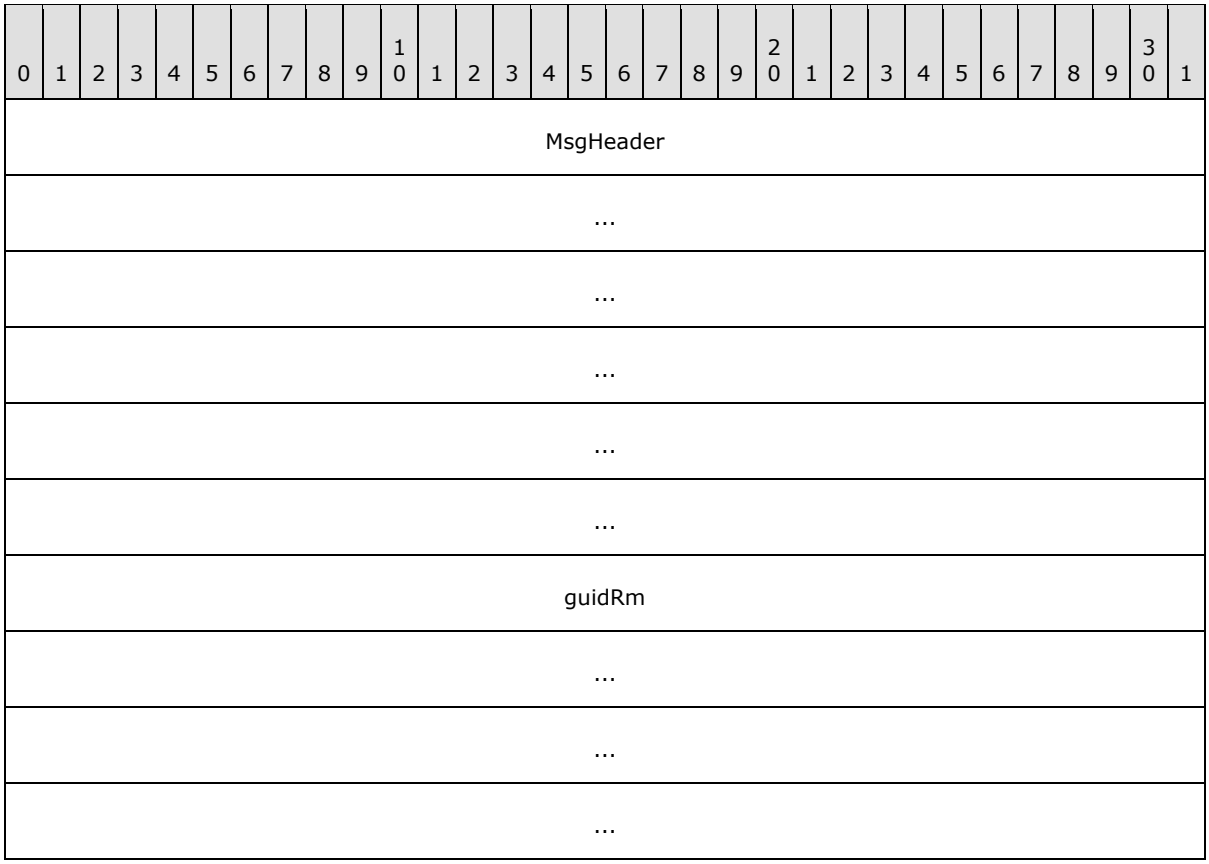
This connection type is used to create an XA Superior Transaction Manager representation in a Transaction Manager and to read from the Transaction Manager the list of XA transactions to be recovered.

For more information about CONNTYPE_XAUSER_CONTROL as an initiator and as an acceptor, see section 3.

This connection type uses the following message types:

2.2.4.2.1 XAUSER_CONTROL_MTAG_CREATE

This message creates an XA Superior Transaction Manager representation in a Transaction Manager.



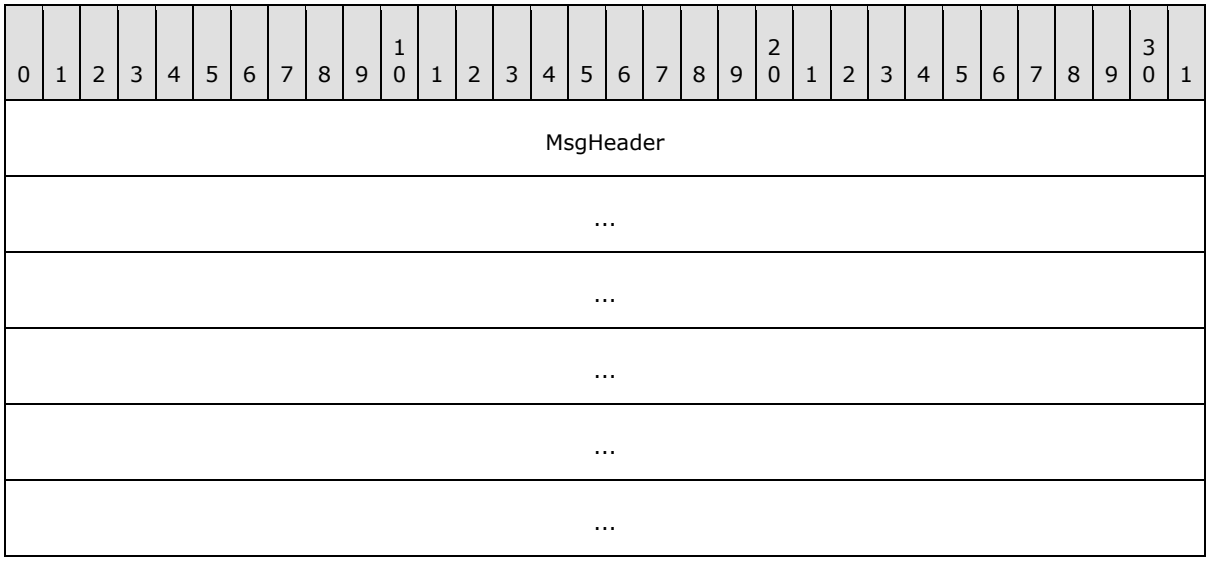
MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004001.
- The **dwcbVarLenData** field MUST be 16.

guidRm (16 bytes): This field MUST contain a **GUID** that specifies an **XA Superior Transaction Manager Identifier** for the XA Superior Transaction Manager.

2.2.4.2.2 XAUSER_CONTROL_MTAG_CREATE_NO_MEM

This message indicates that the request to create an XA Superior Transaction Manager representation failed due to lack of memory.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004006.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.2.3 XAUSER_CONTROL_MTAG_CREATED

This message indicates that the request to create an XA Superior Transaction Manager representation has completed successfully.

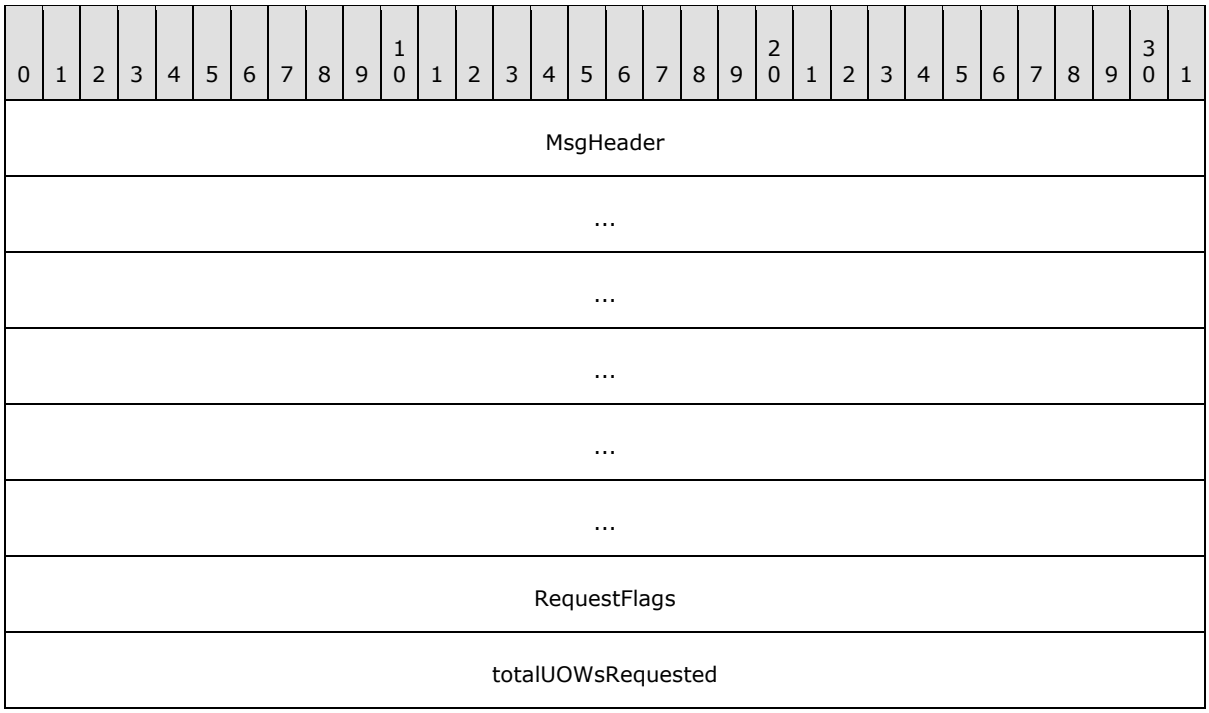
0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004002.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.2.4 XAUSER_CONTROL_MTAG_RECOVER

This message discovers which XA transactions need recovery.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004003.
- The **dwcbVarLenData** field MUST be 8.

RequestFlags (4 bytes): Flags that control the scan performed by the Transaction Manager for XA transactions in need of recovery. It MUST be a bitwise OR operator of one or more the following flags.

Value	Meaning
XARECOVER_START_SCAN 0x00000001	The Transaction Manager MUST begin a new session to scan for transactions in need of recovery.
XARECOVER_END_SCAN 0x00000002	The Transaction Manager MUST end the current session being used to scan for transactions in need of recovery.
XARECOVER_CONTINUE_SCAN 0x00000004	The Transaction Manager MUST continue the current session being used to scan for transactions in need of recovery. This flag is assumed to be set if no other flag is set.

totalUOWsRequested (4 bytes): A 32-bit unsigned integer that MUST contain the requested number of XA transactions that need recovery. This value SHOULD be less than or equal to 5.<6><7><8>

2.2.4.2.5 XAUSER_CONTROL_MTAG_RECOVER_NO_MEM

This message indicates the recovery request failed due to lack of memory.

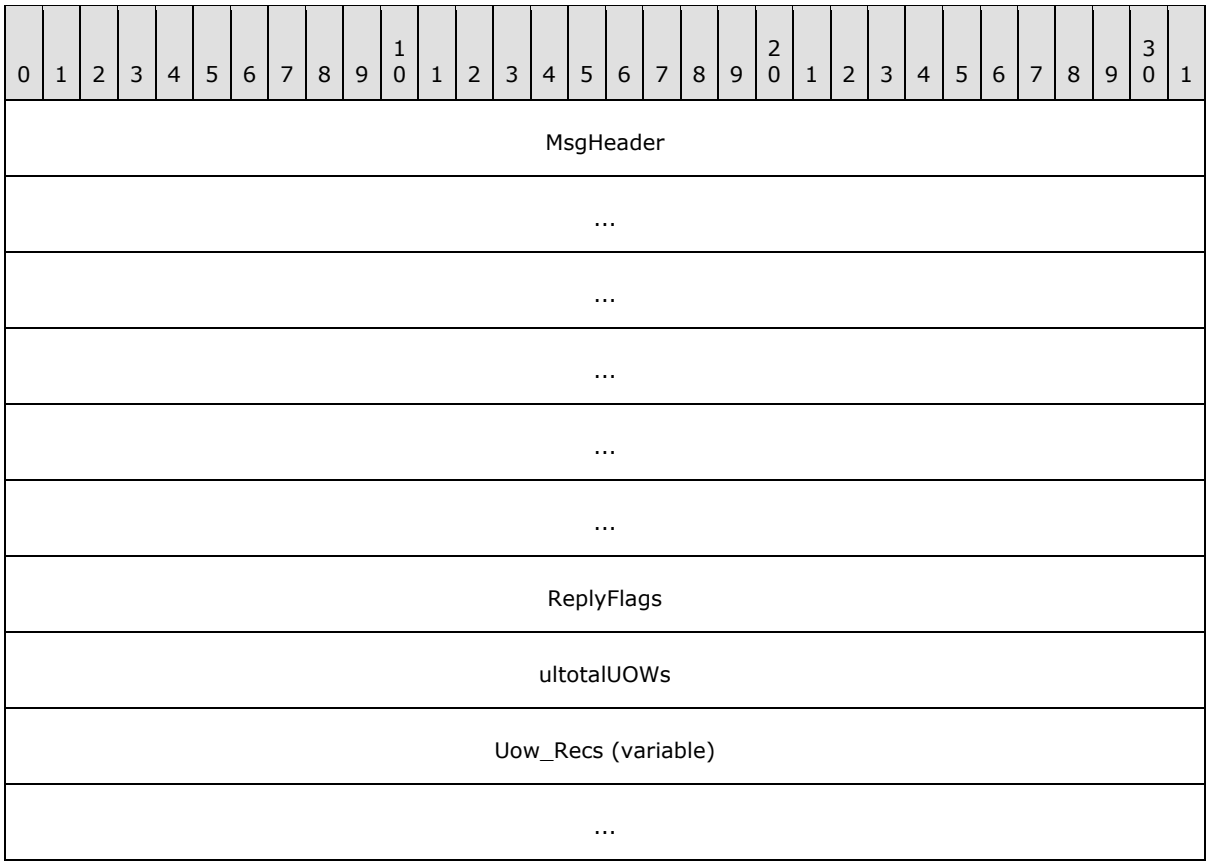
0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004004.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.2.6 XAUSER_CONTROL_MTAG_RECOVER_REPLY

This message indicates that the recovery request has completed successfully.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004005.
- The **dwcbVarLenData** field MUST be 8 bytes plus the number of bytes in the **Uow_Recs** array.

ReplyFlags (4 bytes): Flags that indicate whether more unrecovered transactions remain. It MUST be one of the following flags.

Value	Meaning
XARECOVER_MORE_TO_COME 0x00000001	Not all recovered transactions have been scanned.
XARECOVER_END_OF_RECS 0x00000002	All recovered transactions have been scanned.

ultotalUOWs (4 bytes): A 32-bit unsigned integer that MUST contain the count of elements in the Uow_Recs array, each of which represents an XA transaction in need of recovery. This value SHOULD match the totalUOWsRequested value in the previous XAUSER_CONTROL_MTAG_RECOVER message, unless the scan has completed, in which case the **ultotalUOWsvalue** MIGHT be less.<9>

Uow_Recs (variable): An array of XA_UOW structures. The number of elements in the array MUST be equal to **ultotalUOWs** plus 5. The last 5 elements are reserved, MUST be set to implementation-specific values, and MUST be ignored on receipt.

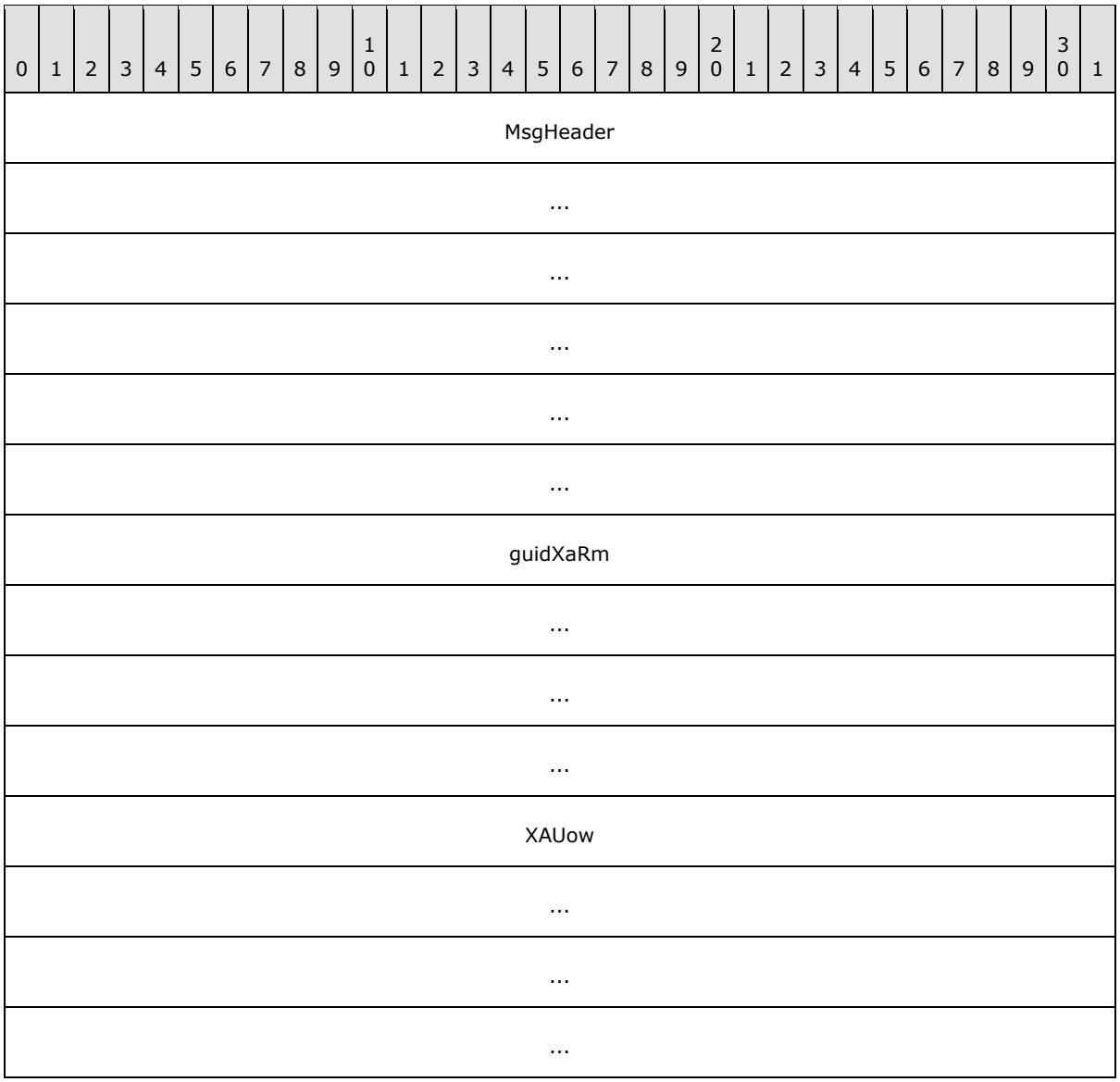
2.2.4.3 CONNTYPE_XAUSER_XACT_START

This connection type is used to start a **loosely-coupled** XA Transaction Branch.

For more information about CONNTYPE_XAUSER_XACT_START as an initiator and as an acceptor, see section 3.

2.2.4.3.1 XAUSER_XACT_MTAG_START

This message creates an XA Transaction Branch.



...
...
...
...
(XAUow cont'd for 28 rows)
isoLevel (optional)
Timeout (optional)
szDesc (optional)
...
...
...
...
...
...
...
(szDesc (optional) cont'd for 2 rows)
isoFlags (optional)

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004010.
- The **dwcbVarLenData** field MUST be one of the following:

Value	Meaning
160	The **isoLevel**, **dwTimeout**, **szDesc**, and **isoFlags** fields are not present in the

Value	Meaning
	message.
212	The **isoLevel**, **dwTimeout**, **szDesc**, and **isoFlags** fields are present in the message.

guidXaRm (16 bytes): This field MUST contain a **GUID** that specifies an XA Superior Transaction Manager Identifier for the XA Superior Transaction Manager.

XAUow (144 bytes): This field MUST contain an XA_UOW structure that specifies a length-qualified XA Transaction Branch Identifier.

isoLevel (4 bytes): This field MUST contain one value from the OLETX_ISOLATION_LEVEL enumeration (as specified in [MS-DTCO]) that specifies the isolation level of the transaction. It is present only when the **dwcbVarLenData** field is set to 212.

Timeout (4 bytes): A 32-bit unsigned integer value that MUST contain the time-out value, in milliseconds, for the transaction. A value of zero MUST be interpreted as an infinite time-out. A transaction SHOULD NOT abort due to time-out before the time-out that is specified by this value has expired. This field is present only when the **dwcbVarLenData** field is set to 212.

szDesc (40 bytes): The description of the transaction, as a fixed-size array of 40 bytes that contains a null-terminated Latin-1 ANSI string, as specified in [ISO-8859-1]. This field MUST be set to an implementation-specific value. Any bytes that follow the first null-terminator character SHOULD be set to zero, and MUST<10> be ignored on receipt. It is present only when the **dwcbVarLenData** field is set to 212.

isoFlags (4 bytes): The isolation flags for the transaction. This field MUST contain the bitwise OR operator of zero or more values from the OLETX_ISOLATION_FLAGS enumeration (as specified in [MS-DTCO]). It is present only when the **dwcbVarLenData** field is set to 212.

2.2.4.3.2 XAUSER_XACT_MTAG_START_DUPLICATE

This message indicates that the request to create an XA Transaction Branch failed because an XA Transaction Branch with the same XA Transaction Branch Identifier already exists.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004021.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.3.3 XAUSER_XACT_MTAG_START_LOG_FULL

This message indicates that the request to create an XA Transaction Branch failed because of lack of log space.

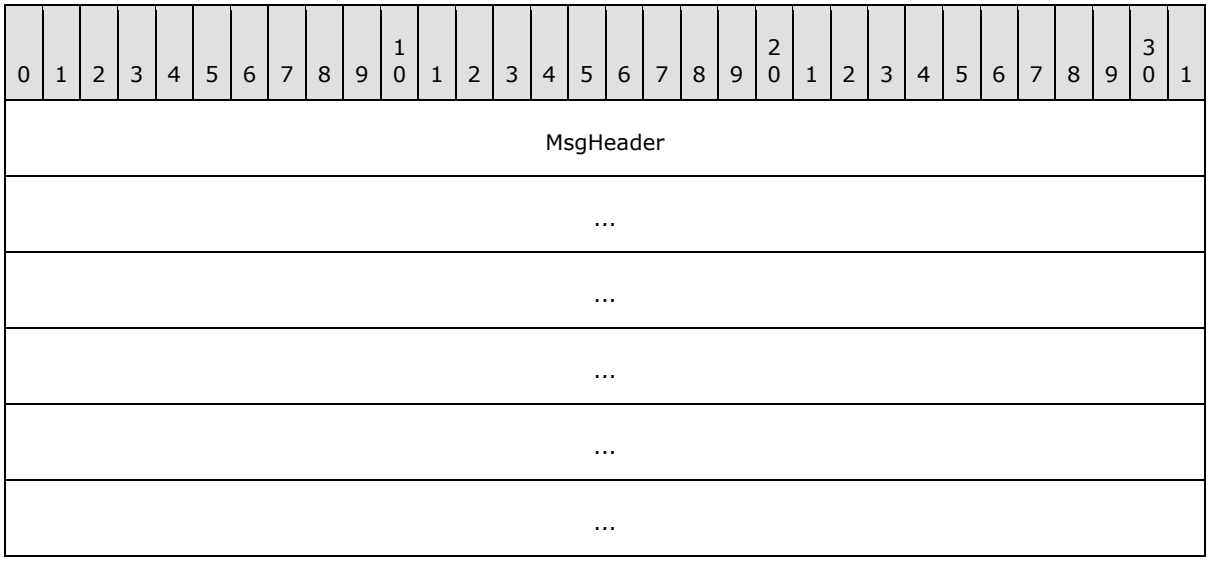
0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004020.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.3.4 XAUSER_XACT_MTAG_START_NO_MEM

This message indicates that the request to create an XA Transaction Branch failed because of lack of memory.



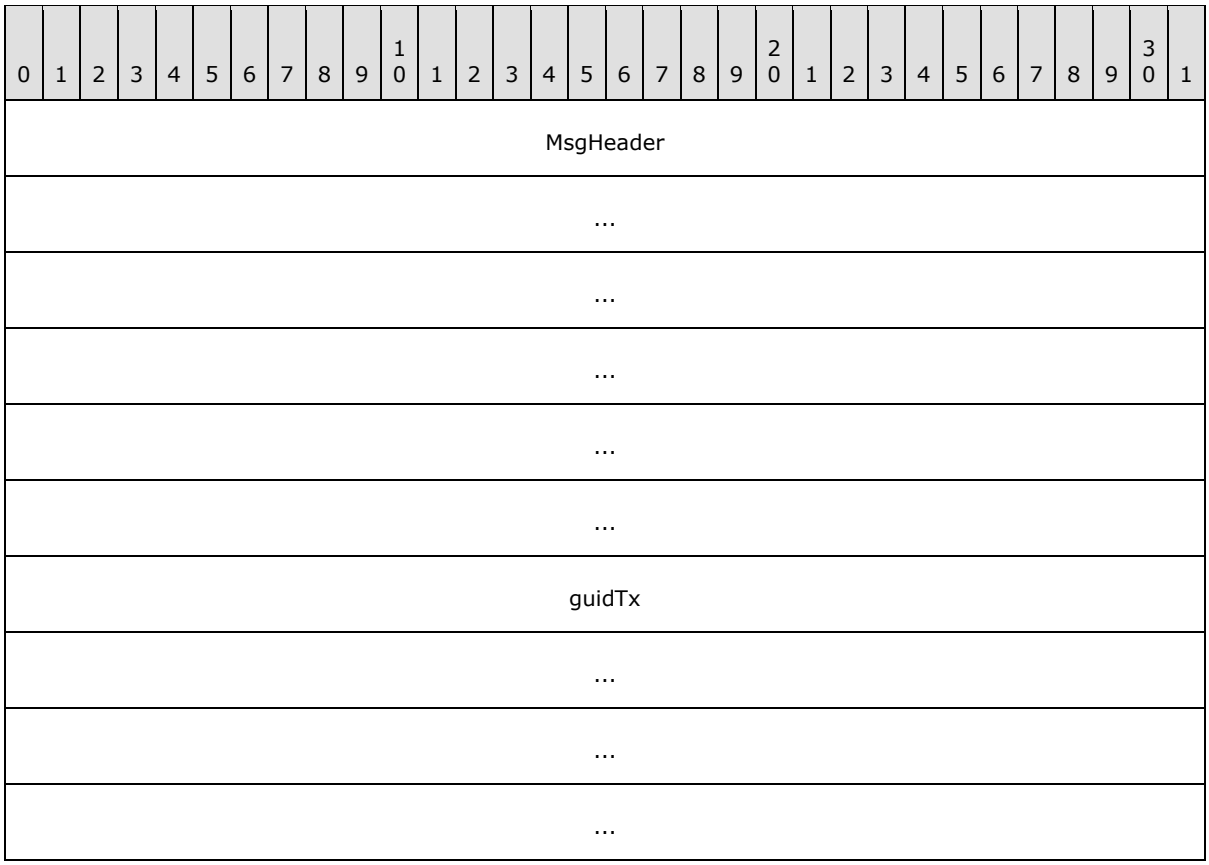
MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004019.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.3.5 XAUSER_XACT_MTAG_STARTED

This message indicates that the request to create an XA Transaction Branch has completed successfully.

:



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004011.
- The **dwcbVarLenData** field MUST be 16.

guidTx (16 bytes): This field MUST contain a **GUID** that specifies the transaction identifier.

2.2.4.4 CONNTYPE_XAUSER_XACT_BRANCH_START

This connection type is used to start a **tightly-coupled** XA Transaction Branch.

For more information about CONNTYPE_XAUSER_XACT_BRANCH_START as an initiator, see section 3.3.5.5, and as an acceptor, see section 3.2.5.5.

The same set of message types in section 2.2.4.3 are sent on this connection type.

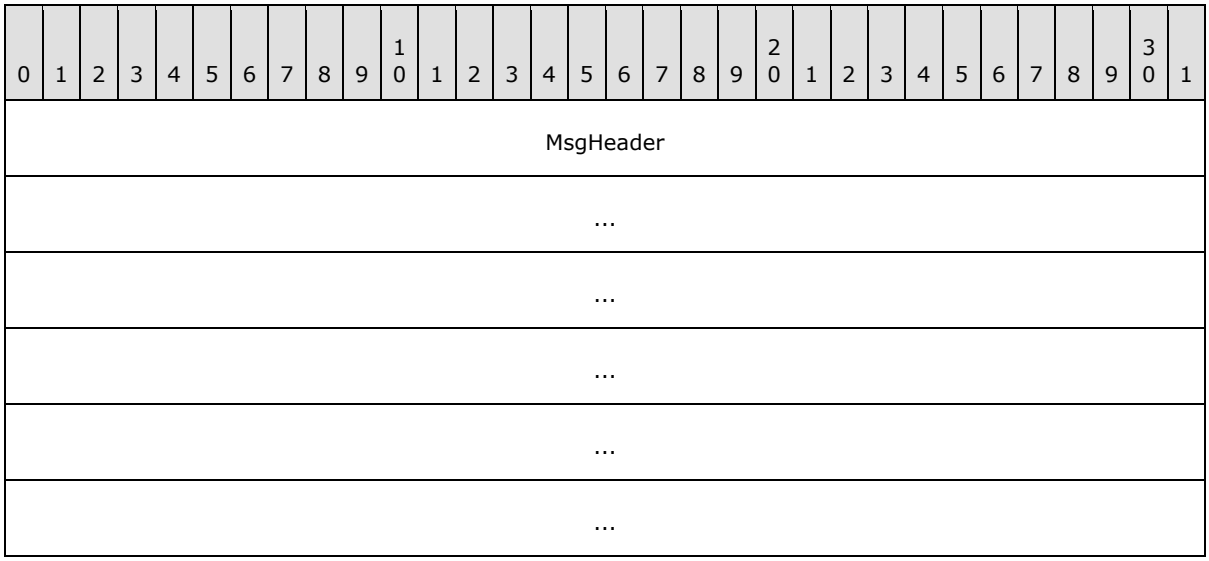
2.2.4.5 CONNTYPE_XAUSER_XACT_OPEN

This connection type is used to open a loosely-coupled XA Transaction Branch for exchanging transactional information.

For more details on CONNTYPE_XAUSER_XACT_OPEN as an initiator and as an acceptor, see section 3

2.2.4.5.1 XAUSER_XACT_MTAG_ABORT

This message requests an attempt to abort the XA Transaction Branch.

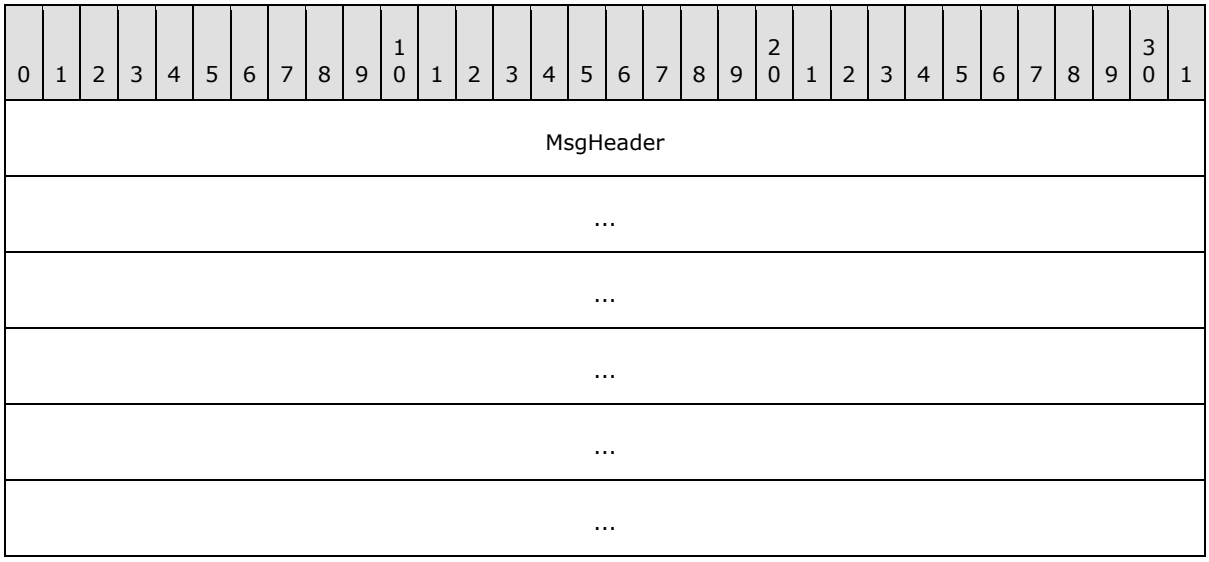


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004014.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.5.2 XAUSER_XACT_MTAG_COMMIT

This message requests an attempt to commit the XA Transaction Branch.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004016.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.5.3 XAUSER_XACT_MTAG_OPEN

This message opens an XA Transaction Branch to exchange transactional information.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															
guidXaRm																															
XAUow																															
...																															
...																															
...																															
...																															
...																															
...																															
...																															
(XAUow cont'd for 28 rows)																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

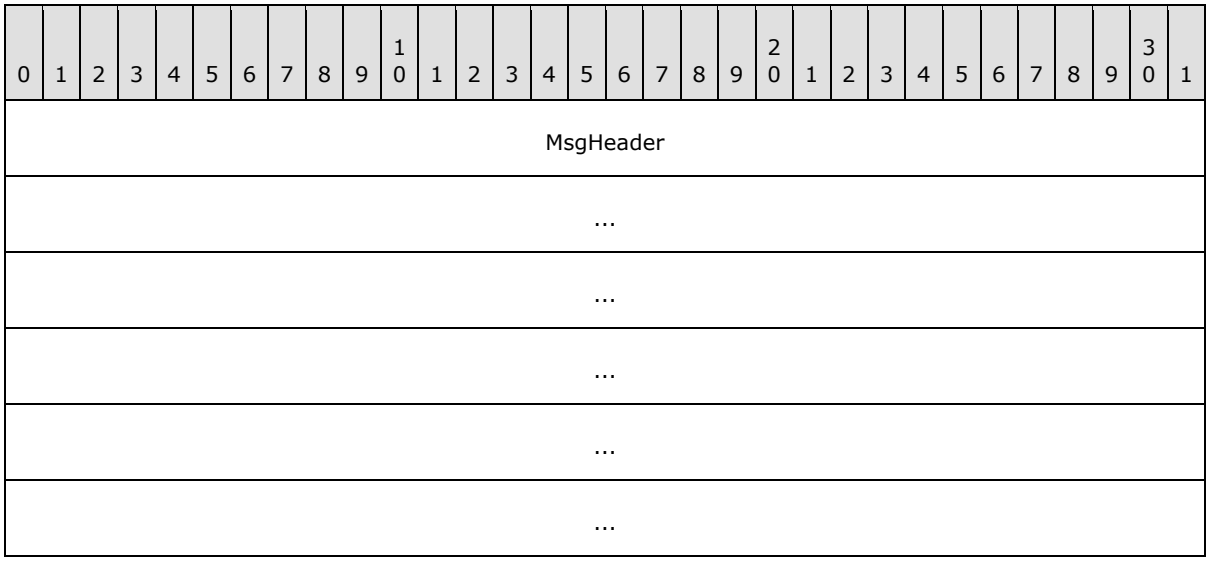
- The **dwUserMsgType** field MUST be 0x00004012.
- The **dwcbVarLenData** field MUST be 160.

guidXaRm (4 bytes): This field MUST be a **GUID** that specifies an XA Superior Transaction Manager Identifier for the XA Superior Transaction Manager.

XAUow (144 bytes): This field MUST contain an XA_UOW structure that specifies a length-qualified XA Transaction Branch Identifier.

2.2.4.5.4 XAUSER_XACT_MTAG_OPEN_NOT_FOUND

This message indicates that the request to open an XA Transaction Branch failed because the transaction could not be found.

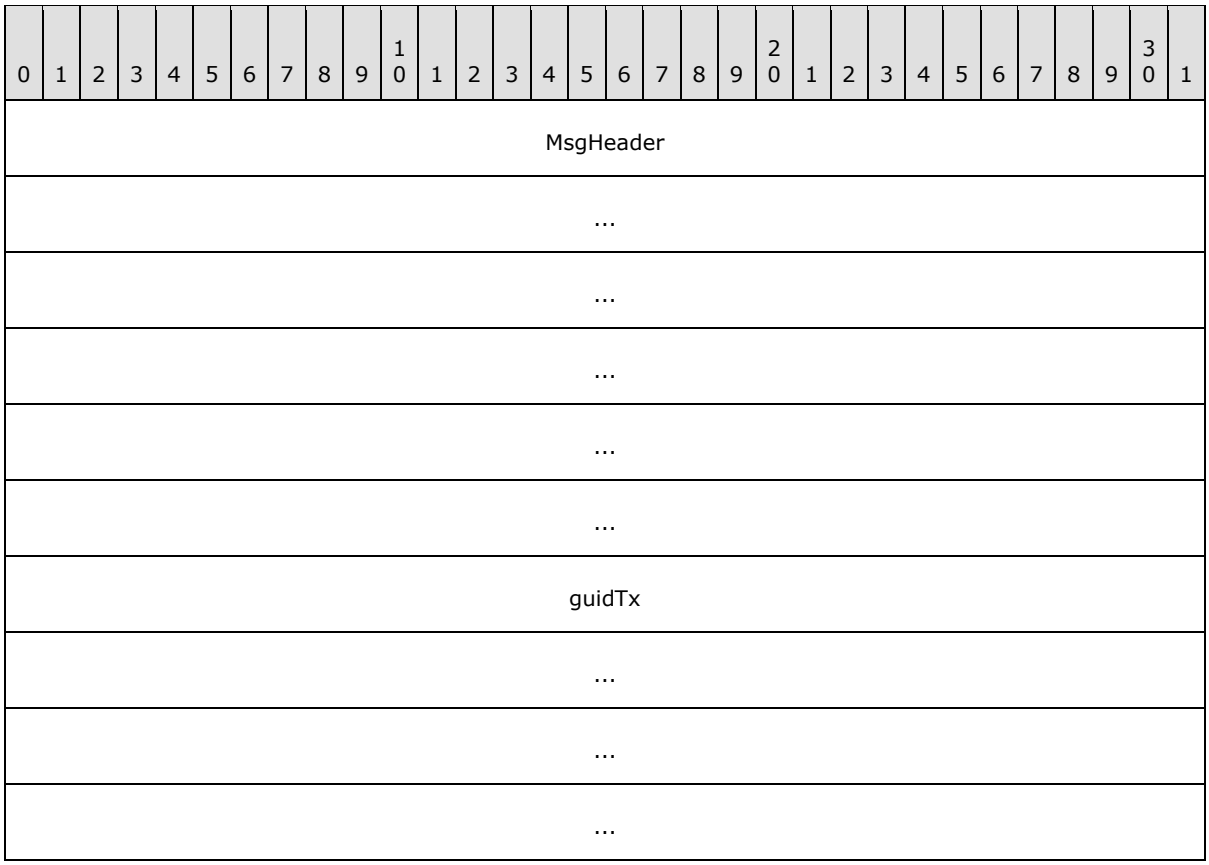


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004022.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.5.5 XAUSER_XACT_MTAG_OPENED

This message indicates that the request to open an XA Transaction Branch has completed successfully.



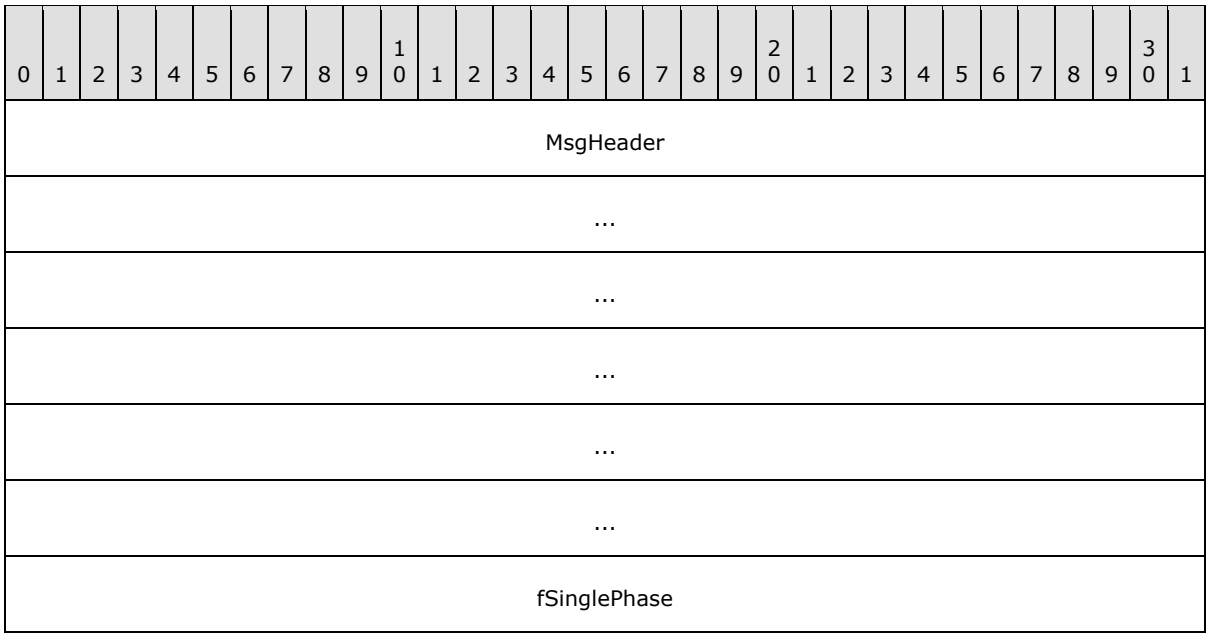
MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004013.
- The **dwcbVarLenData** field MUST be 16.

guidTx (16 bytes): This field MUST contain a **GUID** that specifies the transaction identifier.

2.2.4.5.6 XAUSER_XACT_MTAG_PREPARE

This message requests an attempt to prepare the XA Transaction Branch.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004015.
- The **dwcbVarLenData** field MUST be 4.

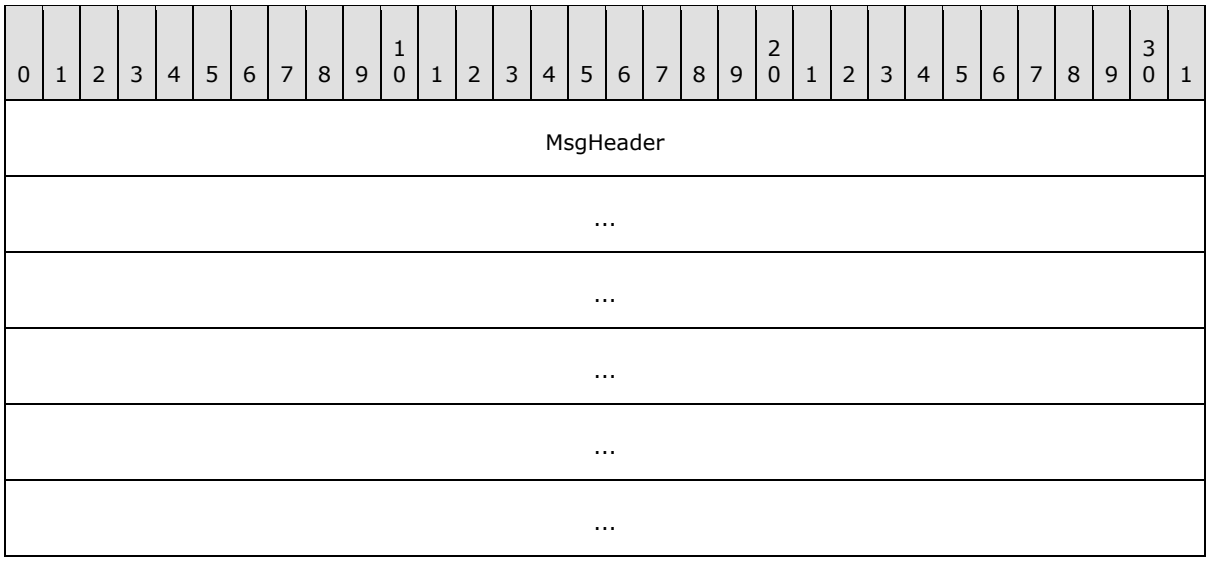
fSinglePhase (4 bytes): A value that indicates whether the XA Superior Transaction Manager wants to do Single-Phase Commit.

It MUST be one of the following values:

Value	Meaning
0x00000000	Single-Phase Commit is not allowed.
0x00000001	Single-Phase Commit is allowed. The transaction manager can do a Single-Phase Commit of the XA Transaction Branch.

2.2.4.5.7 XAUSER_XACT_MTAG_PREPARE_ABORT

This message indicates that the Transaction Manager is preparing to abort the XA Transaction Branch.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004023.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.5.8 XAUSER_XACT_MTAG_PREPARE_SINGLEPHASE_INDOUBT

This message indicates that an attempt was made to do a Single-Phase Commit, but there was a failure that has prevented the Transaction Manager from determining the outcome of the XA Transaction Branch.

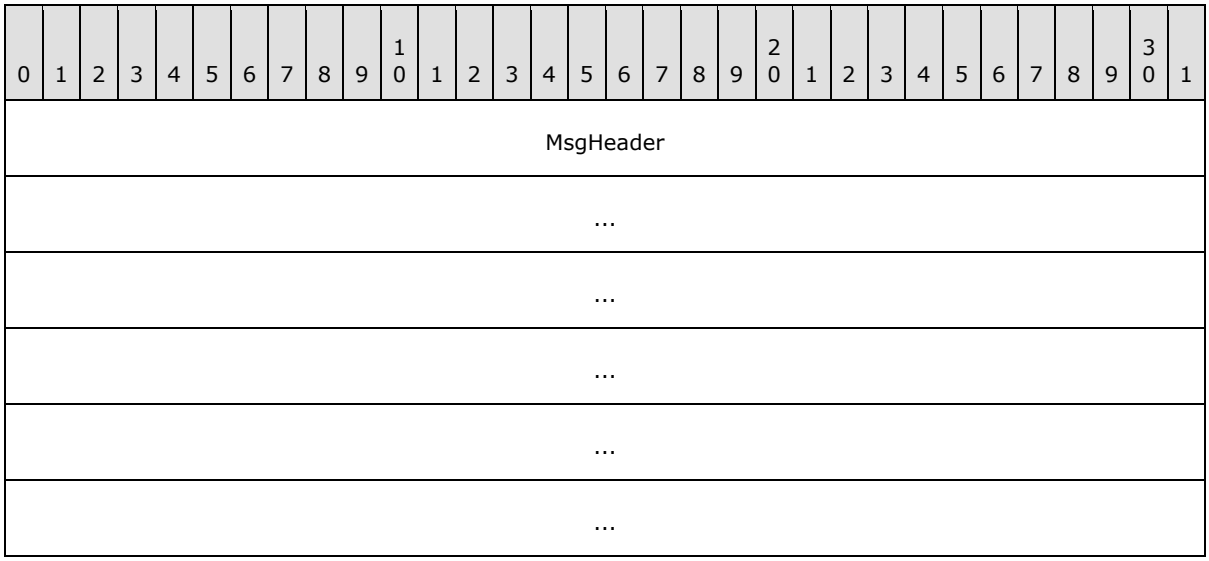
MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004024.

- The **dwcbVarLenData** field MUST be 0.

2.2.4.5.9 XAUSER_XACT_MTAG_REQUEST_COMPLETED

This message indicates that the previous request was completed successfully.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004017.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.5.10 XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL

This message indicates that the previous request failed because of an error in the protocol.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MsgHeader																															
...																															
...																															
...																															
...																															
...																															

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004018.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.6 CONNTYPE_XAUSER_XACT_BRANCH_OPEN

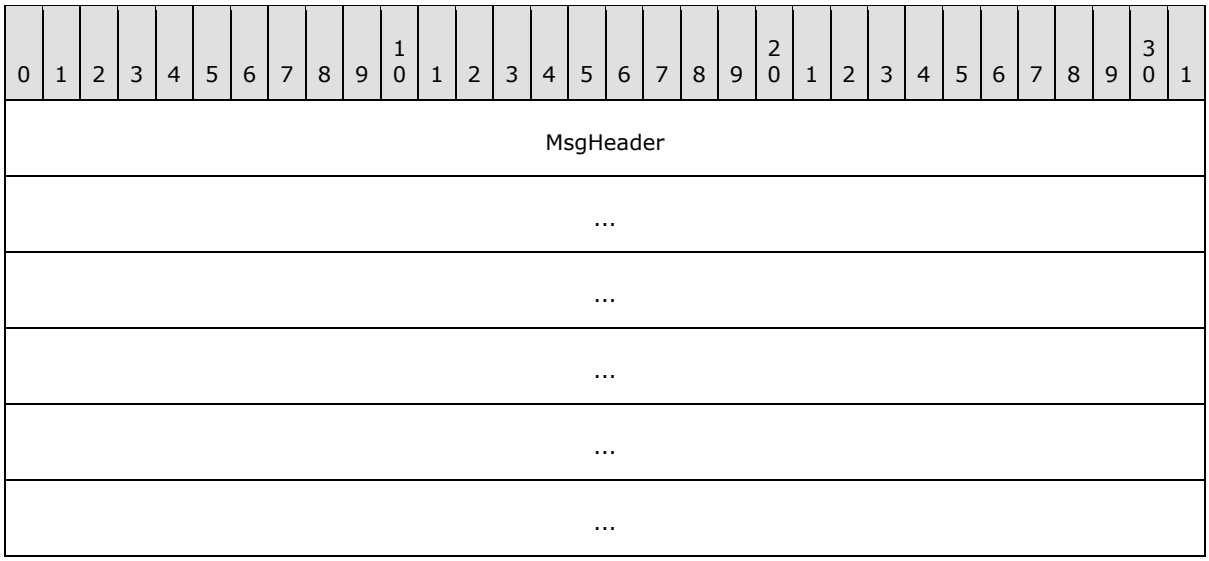
This connection type is used to open a tightly-coupled XA Transaction Branch to exchange transactional information.

XAUSER_XACT_MTAG_READONLY is sent on this connection type, in addition to the set of message types in section 2.2.4.5.

For more information about CONNTYPE_XAUSER_XACT_BRANCH_OPEN as an initiator and as an acceptor, see section 3.

2.2.4.6.1 XAUSER_XACT_MTAG_READONLY

This message indicates that the request to prepare the XA Transaction Branch for commitment was successful and no further involvement in the XA Transaction Branch is required.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004030.
- The **dwcbVarLenData** field MUST be 0.

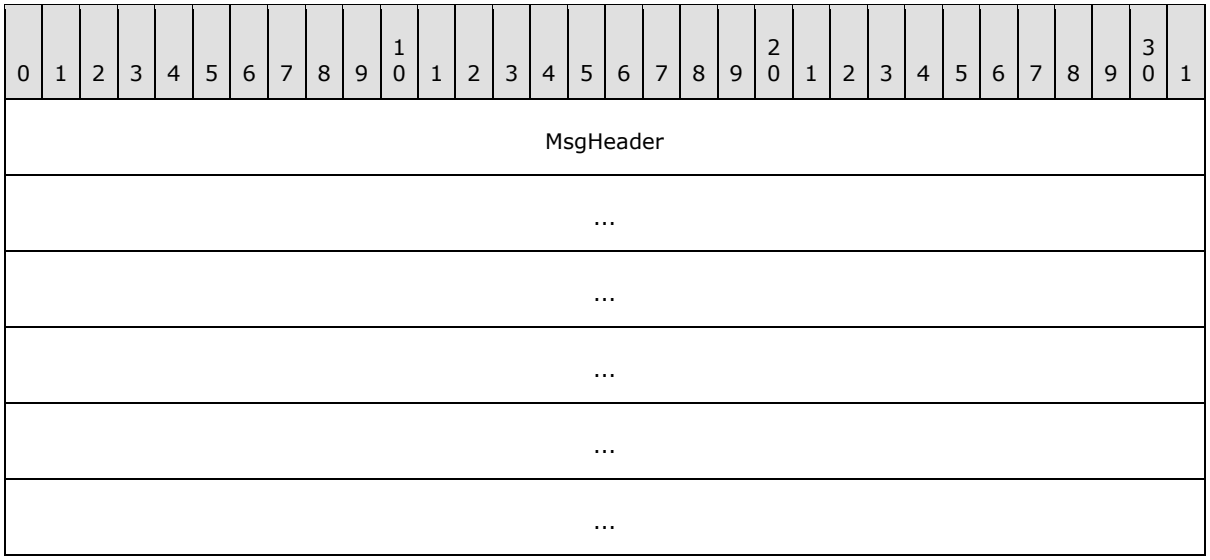
2.2.4.7 CONNTYPE_XAUSER_XACT_MIGRATE

This connection type is used to migrate an XA Transaction Branch between threads of control.

For more information about CONNTYPE_XAUSER_XACT_MIGRATE as an initiator and as an acceptor, see section 3.

2.2.4.7.1 XAUSER_XACT_MTAG_RESUME

This message resumes an XA Transaction Branch on a thread of control.



lenXAIdentifier	padding
guidXaRm	
...	
...	
...	
XAIdentifier	
...	
...	
...	
...	
...	
...	
...	
(XAIdentifier cont'd for 27 rows)	
dwProcessID	
dwThreadID	

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004027.
- The **dwcbVarLenData** field MUST be 168.

lenXAIdentifier (1 byte): An 8-bit unsigned value that MUST contain the length in bytes of the **XAIdentifier** field. The end of this field MUST be padded to the next 4-byte boundary.

padding (3 bytes): Padding to the next 4-byte boundary. Each padding byte MUST be set to an implementation-specific value, and MUST be ignored on receipt.

guidXaRm (16 bytes): This field MUST contain a **GUID** that specifies an XA Superior Transaction Manager Identifier for the XA Superior Transaction Manager.

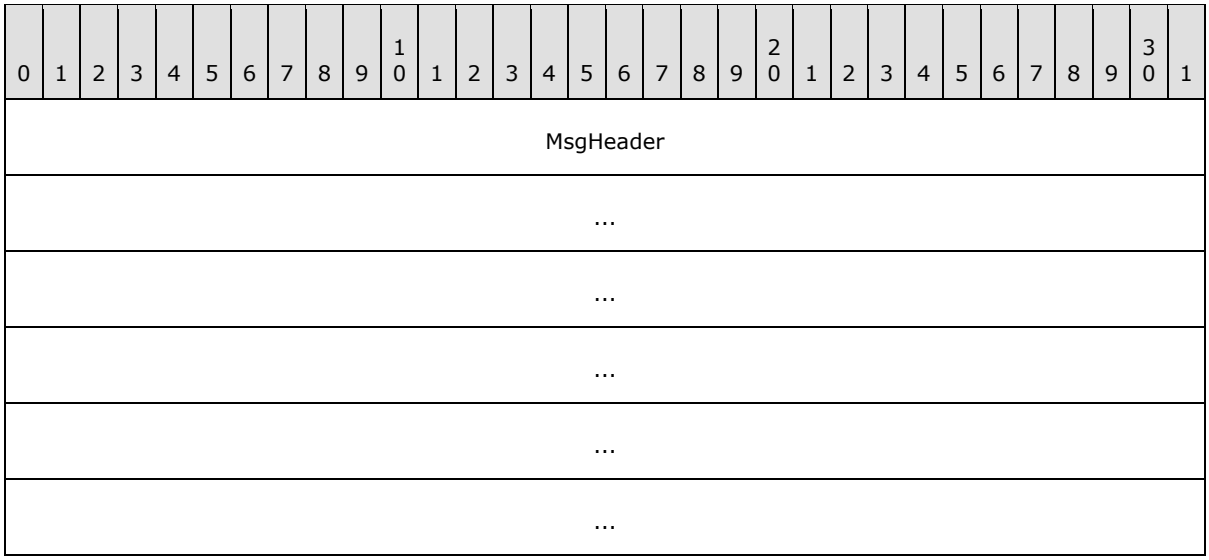
XAIdentifier (140 bytes): This field MUST contain an XA_XID structure that specifies an XA Transaction Branch Identifier.

dwProcessID (4 bytes): A 32-bit unsigned integer that MUST specify the process identifier of the process that is sending the message.

dwThreadID (4 bytes): A 32-bit unsigned integer that MUST specify the thread identifier of the thread of control that is sending the message.

2.2.4.7.2 XAUSER_XACT_MTAG_RESUME_DONE

This message indicates that the XA Transaction Branch was successfully resumed.

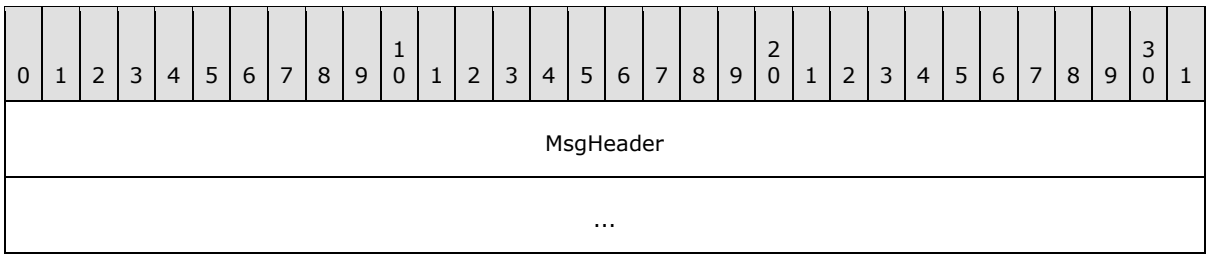


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004028.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.7.3 XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE

This message suspends an XA Transaction Branch on a thread of control for migration.



...	
...	
...	
...	
lenXAIdentifier	padding
guidXaRm	
...	
...	
...	
XAIdentifier	
...	
...	
...	
...	
...	
...	
...	
(XAIdentifier cont'd for 27 rows)	
dwProcessID	
dwThreadID	

MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004025.
- The **dwcbVarLenData** field MUST be 168.

lenXAIdentifier (1 byte): An 8-bit unsigned value that MUST contain the length in bytes of the **XAIdentifier** field. The end of this field MUST be padded to the next 4-byte boundary.

padding (3 bytes): Padding to the next 4-byte boundary. Each padding byte MUST be set to an implementation-specific value, and MUST be ignored on receipt.

guidXaRm (16 bytes): This field MUST contain a **GUID** that specifies an XA Superior Transaction Manager Identifier.

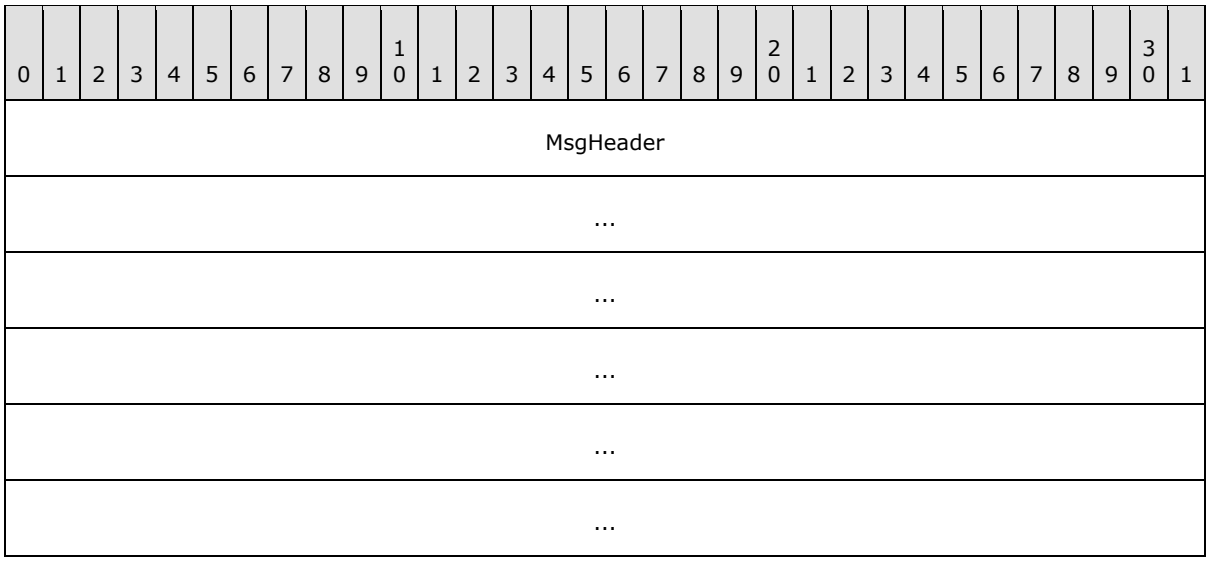
XAIdentifier (140 bytes): This field MUST contain an XA_XID structure that specifies an XA Transaction Branch Identifier.

dwProcessID (4 bytes): A 32-bit unsigned integer that MUST specify the process identifier of the process that sends the message.

dwThreadID (4 bytes): A 32-bit unsigned integer that MUST specify the thread identifier of the thread of control that sends the message.

2.2.4.7.4 XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE

This message indicates that the XA Transaction Branch was successfully suspended for migration.

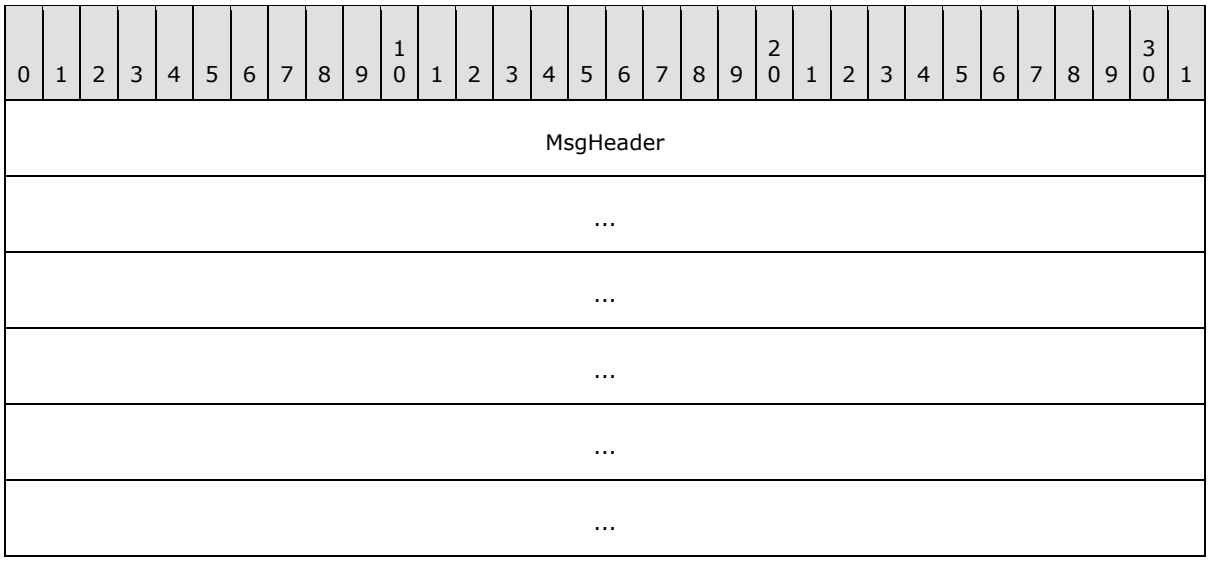


MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004026.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.7.5 XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED

This message indicates that the requested XA Transaction Branch is not currently suspended.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004029.
- The **dwcbVarLenData** field MUST be 0.

2.2.4.8 CONNTYPE_XAUSER_XACT_MIGRATE2

This connection type is used to migrate an XA Transaction Branch between threads of control.

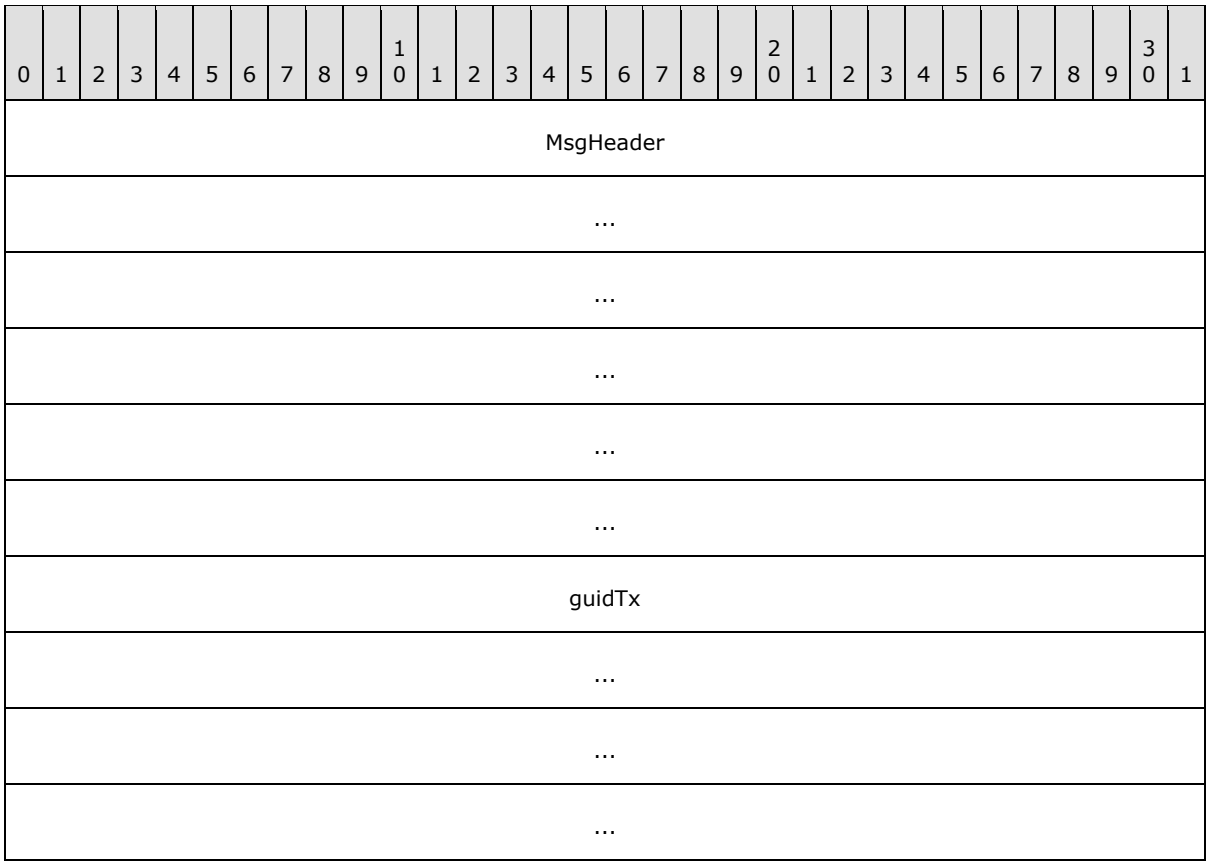
This connection type supersedes CONNTYPE_XAUSER_XACT_MIGRATE.

For more information about CONNTYPE_XAUSER_XACT_MIGRATE2 as an initiator and as an acceptor, see section 3.

The same set of message types in section 2.2.4.7 are sent on this connection type. The syntax of XAUSER_XACT_MTAG_RESUME_DONE is different to support cross-processes migration, as specified in section 2.2.4.8.1.

2.2.4.8.1 XAUSER_XACT_MTAG_RESUME_DONE

This message indicates that the XA Transaction Branch was successfully resumed.



MsgHeader (24 bytes): This field MUST contain a MESSAGE_PACKET structure.

- The **dwUserMsgType** field MUST be 0x00004028.
- The **dwcbVarLenData** field MUST be 16.

guidTx (16 bytes): This field MUST contain a **GUID** that specifies the transaction identifier.

3 Protocol Details

The following sections specify details of the MSDTC Connection Manager: OleTx XA Protocol, including abstract data models, interface method syntax, and message processing rules.

3.1 Common Details

This section defines common details for the transaction participants, as specified in sections 3.2 through 3.5. Each participant MUST conform to the details as specified in this section.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with the behavior that is described in this document.

Note that the abstract data model can be implemented in a variety of ways. This protocol does not prescribe or advocate any specific implementation technique.

A participant MUST maintain all the data elements as specified in [MS-CMP] section 3.1.1.

Participants MUST use MSDTC Connection Manager: OleTx Multiplexing Protocol [MS-CMP] connections as a transport protocol for sending messages. Section 2.1 defines the mechanisms by which this protocol initializes and makes use of the MSDTC Connection Manager: OleTx Multiplexing Protocol.

A participant MUST extend the definition of a connection object, as specified in [MS-CMP], section 3.1.1.1, to include the following data elements.

- **State**: A state enumeration that represents the current state of the connection.

A state enumeration MUST contain a set of values that represent specific states in a logical state machine. For a connection type, these values represent the different states to which the connection's logical state machine is set during the lifetime of the connection.

When a participant initiates or accepts a connection, the **State** field of the connection MUST be set initially to the Idle state. When the connection is disconnected, the connection state MUST be set to the Ended state.

When an instance of a state machine enters the Ended state, the connection that is associated with the state machine MUST be disconnected, if it is not already disconnected, as specified in [MS-CMP] section 3.1.5.1.

A participant MUST support both initiating and accepting multiple concurrent connections of any type inside the same MSDTC Connection Manager: OleTx Transaction Protocol session or different MSDTC Connection Manager: OleTx Transaction Protocol sessions, as specified in [MS-CMPO]. Consequently, a participant MUST support the existence of multiple instances of a single connection of the same type. A participant MUST also support initiating multiple concurrent sessions to a number of different endpoints.

A participant MUST maintain the following data elements:

XA Transactions Enabled flag: A flag used to disable or enable all functionality of the MSDTC Connection Manager: OleTx XA Protocol.

XA Transaction Manager GUID: A GUID used to uniquely identify an XA Transaction Manager. This value must be maintained through failure and recovery.

3.1.2 Timers

None.

3.1.3 Initialization

Initialization occurs as specified in [MS-DTCO] section 3.1.3.

The XA Transactions Enabled flag MUST be set to a value that is obtained from an implementation-specific source.<11>

The XA Transaction Manager GUID MUST be set to a value that is obtained from an implementation-specific source.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

As specified in [MS-DTCO] section 3.1.5.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

As specified in [MS-DTCO] section 3.1.7.

3.2 XA Subordinate Transaction Manager Facet Details

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate the implementations adhere to this model as long as their external behavior is consistent with the behavior that is described in this document.

Note that the abstract data model can be implemented in a variety of ways. This protocol does not prescribe or advocate any specific implementation technique.

The XA Subordinate Transaction Manager Facet MUST maintain all the data elements that are specified in section 3.1.1.

The XA Subordinate Transaction Manager Facet MUST also maintain the following data elements:

3.2.1.1 Versioning

The XA Subordinate Transaction Manager Facet MUST maintain the data that pertains to the branching functionality only on versions where the connection types

CONNTYPE_XAUSER_XACT_BRANCH_START and CONNTYPE_XAUSER_XACT_BRANCH_OPEN are supported as specified in section 2.2.2 . The following data elements, as specified in 3.2.1.2, are affected.

- Wait For All XA Branch Prepares flag
- XA Superior Enlistment Object
  - Coupling
  - Child Branch Table

The XA Subordinate Transaction Manager Facet MUST maintain the data that pertains to the migrate functionality only on versions where the connection type CONNTYPE_XAUSER_XACT_MIGRATE or CONNTYPE_XAUSER_XACT_MIGRATE2 is supported ,as specified in section 2.2.2. The following data elements, as specified in section 3.2.1.2, are affected:

- XA Superior Enlistment Object
  - State
    - Migrate
- XA Child Branch Enlistment Object
  - State
    - Migrate

3.2.1.2 Data Elements

- **Wait For All XA Branch Prepares**: A flag that indicates whether the primary branch of a set of tightly coupled XA Transaction Branches should wait for all branches to receive Prepare requests before initiating Two-Phase Commit processing.
- **XA Superior Table:** A table of XA Superior Objects keyed by Resource Manager Recovery GUID.
- **XA Superior Object**: This object represents an XA Superior Transaction Manager associated with a Resource Manager Recovery GUID. An XA Superior Object MUST contain the following elements:
  - **Open Count**: Specifies how many times xa_open() has been called.
  - **XA Superior Enlistment Table**: A table of XA Superior Enlistment Objects keyed by XID. This table is used to track XA Transactions associated with the Resource Manager Recovery GUID.
  - **Current Recovery XID**: Used as a marker of current XID to be recovered.
- **XA Superior Enlistment Object**: Specifies an XA Superior Enlistment which associates the Primary Branch of an XA Transaction with a transaction Object. This extends the Enlistment Object defined in [MS-DTCO]. An XA Superior Enlistment Object MUST contain the following elements.
  - **Coupling**: Indicates the type of coupling between the XA Transaction Branches. This field MUST be set to one of the following values:

        - Tight: Indicates that different XIDs with same GTRID will be associated with a single Ole Transaction. Also different XIDs that have different GTRIDs will be associated with their own, different Ole Transactions.
        - Loose: Indicates that different XIDS will be associated with their own, different Ole Transactions.
    - Resource Manager Recovery GUID: Associates an Enlistment with XA Superior Transaction Manager.
    - XID: Specifies the XID associated with the XA Superior Enlistment Object.
    - **State**: Specifies the current state of the Enlistment. This field MUST contain one of the following values.
        - **Init:** XA Superior Enlistment Object is not enlisted in a transaction Object.
        - **Active:** XA Superior Enlistment Object is enlisted in a transaction Object.
        - **Migrate:** XA Superior Enlistment Object has been suspended, and must be resumed before it can begin Two-Phase Commit processing.
        - **Preparing:** XA Superior Enlistment Object is awaiting a response to its Prepare request to the associated Transaction Object.
        - **Preparing Single Phase:** XA Superior Enlistment Object is awaiting a response to its Single-Phase Commit to the associated Transaction Object.
        - **Prepared:** State field of the Transaction Object associated with the XA Superior Enlistment Object is set to Prepared.
        - **In Doubt:** State field of the Transaction Object associated with the XA Superior Enlistment Object is set to In Doubt.
        - **Aborting:** XA Superior Enlistment Object is awaiting a response to its abort request to the associated Transaction Object.
        - **Aborted:** State field of the Transaction Object associated with the XA Superior Enlistment Object is set to Aborted.
        - **Committing:** XA Superior Enlistment Object is awaiting a response to its commit request to the associated Transaction Object.
        - **Committed:** State field of the Transaction Object associated with the XA Superior Enlistment Object is set to Committed.
    - **Current Request CMP Connection**: Specifies the CMP Connection associated with the XA Superior Transaction Manager request being processed.
    - **Child Branch table**: A table of XA Superior Child Branch Enlistment Objects keyed by XID.
- **XA Superior Child Branch Enlistment Object**: Specifies an XA Enlistment which associates the **Child Branch** of an XA Transaction with the Primary Branch. This Object MUST contain the following elements.
    - XID: Specifies the XID associated with the XA Superior Enlistment Object.

- **State**: Specifies the current state of the Enlistment. This field MUST contain one of the following values.
  - **Init**: This is the initial state.
  - **Active**: The request received from the XA Superior Transaction Manager is being processed.
  - Child Branch: The request to suspend a transaction has been processed.
  - **Complete**: This is the final state.
- **Parent XA Superior Enlistment Object**: Specifies the XA Superior Enlistment Object associated with the **Parent Branch**.
- **Current Request CMP Connection**: Specifies the CMP Connection associated with the XA Superior Transaction Manager request being processed.

- **XA Superior Enlistment CMP Connection**: CMP Connection associated with CONNTYPE_XAUSER_XACT_START, CONNTYPE_XAUSER_XACT_OPEN, CONNTYPE_XAUSER_XACT_MIGRATE, CONNTYPE_XAUSER_XACT_BRANCH_START, CONNTYPE_XAUSER_XACT_BRANCH_OPEN, and CONNTYPE_XAUSER_XACT_MIGRATE2 acceptors. The definition of an [MS-CMP] connection Object is extended to include the following element:
  - Reference to XA Superior Enlistment Object or XA Superior Child Branch Enlistment Object.
- **XA Superior CMP Connection**: CMP Connection associated with CONNTYPE_XAUSER_CONTROL acceptor. The definition of an [MS-CMP] connection Object is extended to include the following element.
  - Reference to XA Superior Object.

3.2.1.3 CONNTYPE_XAUSER_CONTROL Acceptor States

The XA Subordinate Transaction Manager Facet MUST act as an acceptor for the CONNTYPE_XAUSER_CONTROL connection type. In this role, the XA Subordinate Transaction Manager Facet MUST provide support for the following states.

- Idle
- Active
- Ended

3.2.1.3.1 Idle

This is the initial state. The following event is processed in this state.

- Receiving an XAUSER_CONTROL_MTAG_CREATE Message

3.2.1.3.2 Active

The following events are processed in this state.

- Receiving an XAUSER_CONTROL_MTAG_RECOVER Message

- Connection Disconnected
- Connection Down

3.2.1.3.3 Ended

This is the final state.

3.2.1.3.4 State Diagram

The following figure shows the relationship between the CONNTYPE_XA_USER_CONTROL acceptor states.

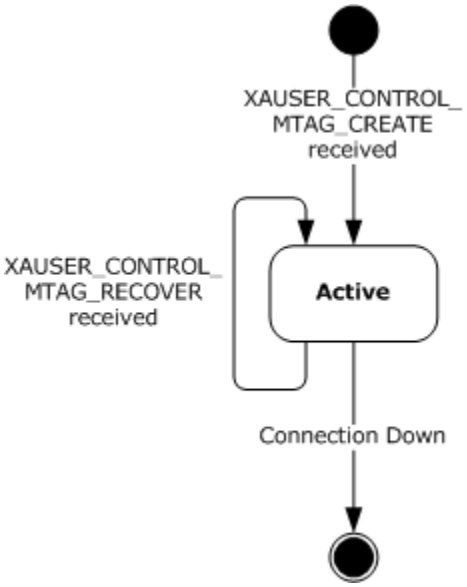


Figure 11: CONNTYPE_XAUSER_CONTROL acceptor states

3.2.1.4 CONNTYPE_ XAUSER_XACT_START Acceptor States

The XA Subordinate Transaction Manager Facet MUST act as an acceptor for the CONNTYPE_XAUSER_XACT_START connection type. In this role, the XA Subordinate Transaction Manager Facet MUST provide support for the following states.

- Idle
- Active
- Ended

3.2.1.4.1 Idle

This is the initial state. The following event is processed in this state.

- Receiving an XAUSER_XACT_MTAG_START Message

3.2.1.4.2 Active

The following events are processed in this state.

- Connection Disconnected
- Connection Down

3.2.1.4.3 Ended

This is the final state.

3.2.1.4.4 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_START acceptor states.

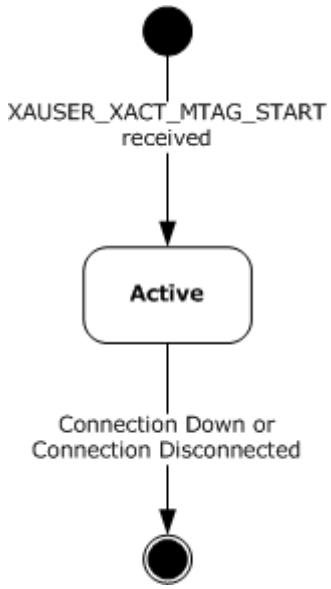


Figure 12: CONNTYPE_XAUSER_XACT_START acceptor states

3.2.1.5 CONNTYPE_ XAUSER_XACT_OPEN Acceptor States

The XA Subordinate Transaction Manager Facet MUST act as an acceptor for the CONNTYPE_XAUSER_XACT_OPEN connection type. In this role, the XA Subordinate Transaction Manager Facet MUST provide support for the following states.

- Idle
- Active
- Ended

3.2.1.5.1 Idle

This is the initial state. The following event is processed in this state.

- Receiving an XAUSER_XACT_MTAG_OPEN Message

3.2.1.5.2 Active

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_PREPARE Message
- Receiving an XAUSER_XACT_MTAG_ABORT Message
- Receiving a XAUSER_XACT_MTAG_COMMIT Message
- Connection Disconnected
- Connection Down

3.2.1.5.3 Ended

This is the final state.

3.2.1.5.4 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_OPEN acceptor states.

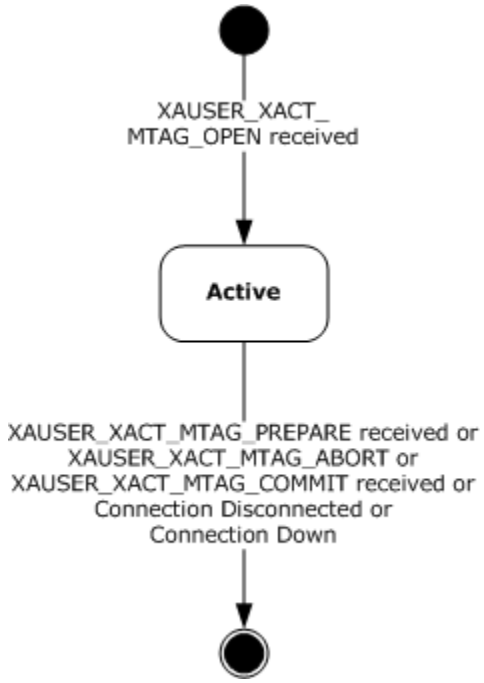


Figure 13: CONNTYPE_XAUSER_XACT_OPEN acceptor states

3.2.1.6 CONNTYPE_ XAUSER_XACT_MIGRATE Acceptor States

The XA Subordinate Transaction Manager Facet MUST act as an acceptor for the CONNTYPE_XAUSER_XACT_MIGRATE connection type. In this role, the XA Subordinate Transaction Manager Facet MUST provide support for the following states.

- Idle
- Ended

3.2.1.6.1 Idle

This is the initial state. The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE Message
- Receiving an XAUSER_XACT_MTAG_RESUME Message

3.2.1.6.2 Ended

This is the final state.

3.2.1.6.3 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_MIGRATE acceptor states.

Figure 14: CONNTYPE_XAUSER_XACT_MIGRATE acceptor states

3.2.1.7 CONNTYPE_ XAUSER_XACT_BRANCH_START Acceptor States

The XA Subordinate Transaction Manager Facet MUST act as an acceptor for the CONNTYPE_XAUSER_XACT_BRANCH_START connection type. In this role, the XA Subordinate Transaction Manager Facet MUST provide support for the following states.

- Idle
- Active
- Ended

3.2.1.7.1 Idle

This is the initial state. The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_START Message.

3.2.1.7.2 Active

The following events are processed in this state.

- Connection Disconnected
- Connection Down

3.2.1.7.3 Ended

This is the final state.

3.2.1.7.4 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_BRANCH_START acceptor states.

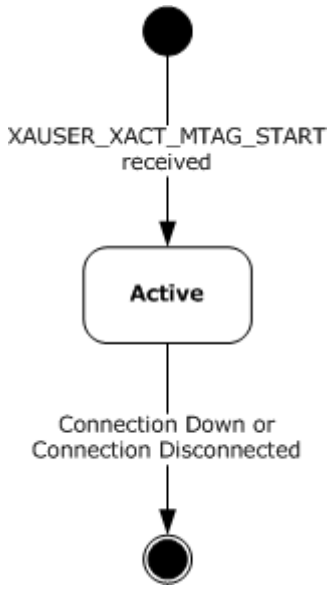


Figure 15: CONNTYPE_XAUSER_XACT_BRANCH_START acceptor states

3.2.1.8 CONNTYPE_ XAUSER_XACT_BRANCH_OPEN Acceptor States

The XA Subordinate Transaction Manager Facet MUST act as an acceptor for the CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection type. In this role, the XA Subordinate Transaction Manager Facet MUST provide support for the following states.

- Idle
- Active
- Ended

3.2.1.8.1 Idle

This is the initial state. The following event is processed in this state.

- Receiving an XAUSER_XACT_MTAG_OPEN Message

3.2.1.8.2 Active

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_PREPARE Message
- Receiving an XAUSER_XACT_MTAG_ABORT Message
- Receiving an XAUSER_XACT_MTAG_COMMIT Message
- Connection Disconnected
- Connection Down

3.2.1.8.3 Ended

This is the final state.

3.2.1.8.4 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_BRANCH_OPEN acceptor states.

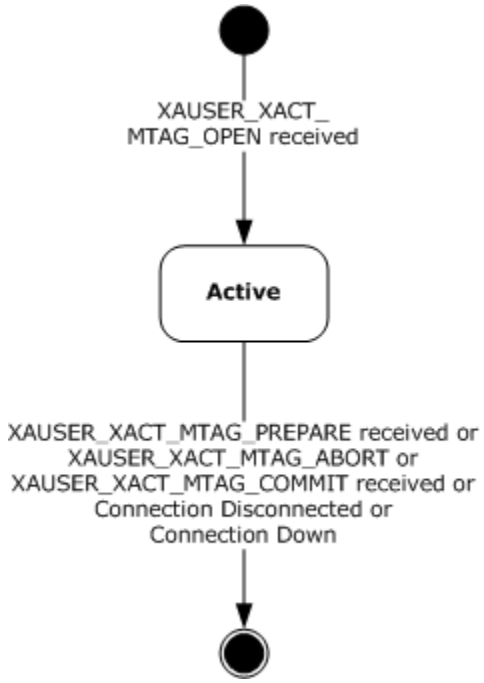


Figure 16: CONNTYPE_ XAUSER_XACT_BRANCH_OPEN acceptor states

3.2.1.9 CONNTYPE_ XAUSER_XACT_MIGRATE2 Acceptor States

The XA Subordinate Transaction Manager Facet MUST act as an acceptor for the CONNTYPE_XAUSER_XACT_MIGRATE2 connection type. In this role, the XA Subordinate Transaction Manager Facet MUST provide support for the following states.

- Idle
- Ended

3.2.1.9.1 Idle

This is the initial state. The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE Message
- Receiving an XAUSER_XACT_MTAG_RESUME Message

3.2.1.9.2 Ended

This is the final state.

3.2.1.9.3 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_MIGRATE2 acceptor states.

Figure 17: CONNTYPE_XAUSER_XACT_MIGRATE2 acceptor states

3.2.1.10 XA Superior Enlistment State Diagram

The following figure shows the relationship between the XA Superior Enlistment states.

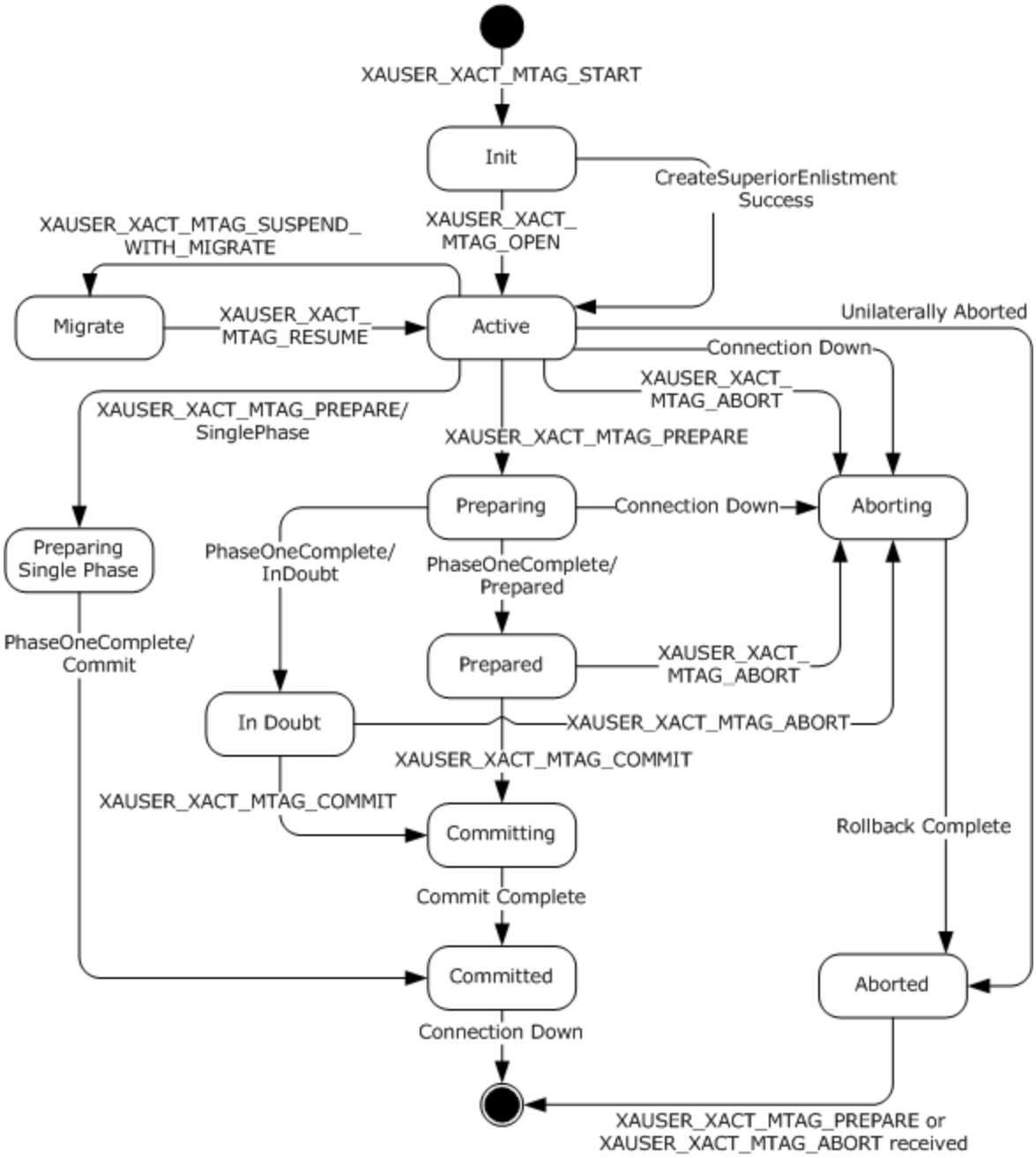


Figure 18: XA Superior Enlistment Object states

3.2.2 Timers

None.

3.2.3 Initialization

When the XA Subordinate Transaction Manager Facet is initialized:

- The Wait For All XA Branch Prepares flag should be set to an implementation-specific value.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 CONNTYPE_XAUSER_CONTROL as Acceptor

This is an XA Superior CMP Connection.

For all messages received in this Connection Type, the XA Subordinate Transaction Manager Facet MUST process the message as specified in section 3.1. The XA Subordinate Transaction Manager Facet MUST also follow the processing rules specified in the following sections.

3.2.5.1.1 Receiving an XAUSER_CONTROL_MTAG_CREATE Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_CONTROL_MTAG_CREATE message, it MUST perform the following actions.

- If the connection state is Idle:
  - Attempt to find the XA Superior Object in the XA Superior Table keyed by the **guidXaRm** field of the message.
  - If an Object is found:
    - Increment the Open Count field of the found XA Superior Object.
    - Send an XAUSER_CONTROL_MTAG_CREATED message using the connection.
    - Set the connection state to Active.
    - Set the XA Superior Object referenced by the XA Superior CMP Connection to the found XA Superior Object.
  - Otherwise:
    - Attempt to create a new XA Superior Object with the following values.
      - Set the Open Count field to 1.
    - If the creation of the new XA Superior Object fails:
      - Send a XAUSER_CONTROL_MTAG_CREATE_NO_MEM message using the connection.
    - Otherwise:
      - Add the new XA Superior Object to the XA Superior Table referenced by the XA Subordinate Transaction Manager Facet with the **guidXaRm** field of the message as its key.
      - Attempt to find the XA Superior Enlistment Objects in the Enlistment Table referenced by the Core Transaction Manager Facet that meet the following condition.
        - Resource Manager Recovery GUID set to the **guidXaRm** field of the message.

- For each XA Superior Enlistment Object found:
  - Add the found XA Superior Enlistment Object to the XA Superior Enlistment Table referenced by the created XA Superior Object.
- Send an XAUSER_CONTROL_MTAG_CREATED message using the receiving connection.
- Set the connection state to Active.
- Set the XA Superior Object referenced by the XA Superior CMP Connection to the newly created XA Superior Object.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.1.2 Receiving an XAUSER_CONTROL_MTAG_RECOVER Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_CONTROL_MTAG_RECOVER message, it MUST perform the following actions.

- SHOULD check the following conditions<12>
  - The connection state is Active.
  - The **totalUOWsRequested** field of the message is less than or equal to an implementation specific value and not equal to 0.<13>
- If the conditions are met:
  - **totalUOWsRequested** MAY be set to the minimum between an implementation specific value and **totalUOWsRequested**<14><15>
  - Create a new XAUSER_CONTROL_RECOVER_REPLY message.
    - The **ReplyFlags** field is set to TMNOFLAGS.
    - The **ultotalUOWs** field is set to 0.
    - The **Uow_Recs** array is set to size of **totalUOWsRequested** field of the received message.
  - If failed:
    - Send a XAUSER_CONTROL_MTAG_RECOVER_NO_MEM message using the receiving connection.
  - Otherwise:
    - If **RequestFlags** field of the message is set to XARECOVER_START_SCAN:
      - Reset the Current Recovery XID field of XA Superior Object referenced by receiving connection.
    - If the Current Recovery XID field of XA Superior Object referenced by receiving connection is not set and XA Superior table is not empty:
      - Set the Current Recovery XID field of XA Superior Object referenced by receiving connection to the XID field of the first XA Superior Enlistment Object in the XA Superior

Enlistment Table referenced by the XA Superior Object referenced by the receiving connection.

- Perform the following steps until one of the following conditions is TRUE.
  - The Current Recovery XID field of the XA Superior Object referenced by the receiving connection is not set.
  - The **ultotalUOWs** field of the XAUSER_CONTROL_RECOVER_REPLY message is equal to **totalUOWsRequested** field of the XAUSER_CONTROL_RECOVER message.
  - SHOULD test if the **State** field of the XA Superior Enlistment Object corresponding to the Current Recovery XID field of the XA Superior Object referenced by the receiving connection in the XA Superior Enlistment Table referenced by the XA Superior Object referenced by the receiving connection meets one of the following conditions<16>
    - The **State** field is set to Prepared.
    - The **State** field is set to In Doubt.
  - If one of the conditions is met:
    - Add the Current Recovery XID field of the XA Superior Object referenced by the receiving connection to the **Uow_Recs** array referenced by the XAUSER_CONTROL_RECOVER_REPLY message.
    - Increment the **ultotalUOWs** field of the XAUSER_CONTROL_RECOVER_REPLY message.
  - If Current Recovery XID field of the XA Superior Object is not the last XID in the XA Superior Enlistment Table referenced by the XA Superior Object:
    - Set Current Recovery XID field of the XA Superior Object referenced by the receiving connection to the next XID in the XA Superior Enlistment Table referenced by the XA Superior Object.
  - Otherwise:
    - Reset Current Recovery XID field of the XA Superior Object.
- Test the following conditions.
  - The Current Recovery XID field of the XA Superior Object is not set.
  - The **RequestFlags** field of the message is set to XARECOVER_END_SCAN.
- If one of the conditions is met:
  - Set the **ReplyFlags** field of the XAUSER_CONTROL_RECOVER_REPLY message to XARECOVER_END_OF_RECS.
- Otherwise:
  - Set the **ReplyFlags** field of the XAUSER_CONTROL_RECOVER_REPLY message to XARECOVER_MORE_TO_COME.
- Send the XAUSER_CONTROL_MTAG_RECOVER_REPLY message using the receiving connection.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.1.3 Connection Disconnected, Connection Down

When a CONNTYPE_XAUSER_CONTROL connection is disconnected, the XA Subordinate Transaction Manager Facet MUST perform the following actions.

- If XA Superior Object reference is set:
  - Decrement the Open Count field of the XA Superior Object referenced by the connection.
  - If Open Count field of the XA Superior Object is set to 0 perform the following actions.
    - Remove XA Superior Object from XA Superior table.
    - For each XA Superior Enlistment Object in the XA Superior Enlistment Table of the XA Superior Object, if the State field is set to Active:
      - Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.
        - The Transaction Object referenced by the XA Superior Enlistment Object.
    - Delete the XA Superior Object.

3.2.5.2 CONNTYPE_XAUSER_XACT_START as Acceptor

This is an XA Superior Enlistment CMP Connection.

For all messages received in this Connection Type, the XA Subordinate Transaction Manager Facet MUST process the message as specified in the section 3.1. The XA Subordinate Transaction Manager Facet MUST also follow the processing rules specified in the following sections:

3.2.5.2.1 Receiving an XAUSER_XACT_MTAG_START Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_START message, it MUST perform the following actions.

- If the State field of the receiving XA Superior Enlistment CMP Connection is set to Idle:
  - Attempt to find the XA Superior Object in the XA Superior Table keyed by the **guidXaRm** field of the message.
  - If an Object is found:
    - Use this as the XA Superior Object in the remainder of the processing of this event.
  - Otherwise:
    - Attempt to create a new XA Superior Object with the following values.
      - Set the Open Count field to 1.
    - If the creation of the new XA Superior Object fails:
      - Send an XAUSER_XACT_MTAG_START_NO_MEM message using the connection.
    - Otherwise:

- Add the new XA Superior Object to the XA Superior Table referenced by the XA Subordinate Transaction Manager Facet with the **guidXaRm** field of the message as its key.
- Attempt to find the XA Superior Enlistment Objects in the Enlistment Table referenced by the Core Transaction Manager Facet that meet the following condition.
  - Resource Manager Recovery GUID set to the **guidXaRm** field of the message.
- For each XA Superior Enlistment Object found:
  - Add the found XA Superior Enlistment Object to the XA Superior Enlistment Table referenced by the created XA Superior Object.
- Use this as the XA Superior Object in the remainder of the processing of this event.

- Attempt to create a new XA Superior Enlistment Object with the following values.
  - The **Coupling** field is set to Loose.
  - The Resource Manager Recovery GUID field is set to **guidXaRm** field of the message.
  - The XID field is set to the **XAIdentifier** field of the message.
  - The **State** field is set to Init.
  - The Current Request CMP connection field is set to the receiving connection.
- If failed:
  - Send an XAUSER_XACT_MTAG_START_NO_MEM message using the connection.
- Otherwise:
  - Attempt to create new transaction with the following settings.
    - The Isolation Level field is set to the **isoLevel** field of the message or 0.
    - The Timeout field is set to the **Timeout** field of the message or 0.
    - The Description field is set to the **szDesc** of the message or "".
    - The Isolation Flags field is set to **isoFlags** field of the message or 0.
  - If failed:
    - Send an XAUSER_XACT_MTAG_START_NO_MEM message using the connection.
  - Otherwise:
    - Attempt to find XA Superior Enlistment Object in XA Superior Enlistment Table referenced by the XA Superior Object that meet both of the following conditions.
      - The XID field is set to the value of the **XAIdentifier** field of the message.
      - The Coupling field is set to Loose.
    - If found:

- Send an XAUSER_XACT_MTAG_START_DUPLICATE message using the connection.

- Otherwise:

  - Set Transaction field of the XA Superior Enlistment Object to newly created transaction.

  - Signal the Create Superior Enlistment event on the Core Transaction Manager Facet with the following argument.

    - The XA Superior Enlistment Object.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.2.2 Connection Disconnected

When a CONNTYPE_XAUSER_XACT_START connection is disconnected, the XA Subordinate Transaction Manager Facet MUST perform the actions as specified in section 3.1.

3.2.5.2.3 Connection Down

The Connection Down event MUST be signaled with the following argument:

- XA Superior Enlistment CMP Connection.

When a CONNTYPE_XAUSER_XACT_START connection is disconnected, the XA Subordinate Transaction Manager Facet SHOULD perform the following actions:<17>

- If the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection is set to either Active or Preparing.

  - Set the State field of XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection to Aborting.

  - If Current Request CMP Connection field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection is set to provided XA Superior Enlistment CMP Connection:

    - Reset Current Request CMP Connection field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection.

  - Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.

    - The Transaction Object, referenced by the XA Superior Enlistment Object, referenced by the provided XA Superior Enlistment CMP Connection.

3.2.5.3 CONNTYPE_XAUSER_XACT_OPEN as Acceptor

This is an XA Superior Enlistment CMP Connection.

For all messages received in this Connection Type, the XA Subordinate Transaction Manager Facet MUST process the message as specified in section 3.1. The XA Subordinate Transaction Manager Facet MUST also follow the processing rules specified in the following sections.

3.2.5.3.1 Receiving an XAUSER_XACT_MTAG_OPEN Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_OPEN message, it MUST perform the following actions.

- If the connection state is Idle:
  - Attempt to find an XA Superior Object in the XA Superior Table keyed by the **guidXaRm** field of the message.
  - If not found:
    - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
  - Otherwise:
    - Attempt to find XA Superior Enlistment Object in XA Superior Enlistment Table referenced by the XA Superior Object that meet both of the following criteria:
      - The XID field is set to the **XAIdentifier** field of the message.
      - Coupling set to Loose.
    - If not found:
      - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
    - Otherwise:
      - Set the state field of receiving connection to **Active**.
      - Set the state field of the XA Superior Enlistment Object to **Active**.
      - Set the XA Superior Enlistment Object referenced by the receiving XA Superior Enlistment CMP Connection Object to the located XA Superior Enlistment Object.
      - Send an XAUSER_XACT_MTAG_OPENED message with the following arguments.
        - Transaction Identifier of the Transaction Object referenced by the provided XA Superior Enlistment Object.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.3.2 Receiving an XAUSER_XACT_MTAG_PREPARE Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_PREPARE message, it MUST perform the following actions.

- If the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to Active:
  - Set Current Request CMP Connection field of XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to receiving connection.
  - If the **Single Phase** field of the message is set to TRUE:
    - Set the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to Preparing Single Phase.

- Otherwise:
  - Set the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to Preparing.
- Signal the Begin Phase Zero event on the Core Transaction Manager Facet with the following argument.
  - The Transaction Object referenced by the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection.

- Otherwise if the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to Aborted:
  - Send an XAUSER_XACT_MTAG_PREPARE_ABORT message using the connection.
  - Remove XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection from the XA Superior Enlistment Table referenced by the XA Superior Object indexed in the XA Superior Table by Resource Manager Recovery GUID field of the XA Superior Enlistment Object.
- Otherwise:
  - Send XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message upon receiving connection.

3.2.5.3.3 Receiving an XAUSER_XACT_MTAG_COMMIT Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_COMMIT message, it MUST perform the following actions.

- If the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to either Prepared or In Doubt:
  - Set the Current Request CMP Connection field of XA Superior Enlistment referenced by the XA Superior Enlistment CMP Connection to receiving connection.
  - Set the State field of XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to Committing.
  - Signal the Begin Commit event on the Core Transaction Manager Facet with the following argument.
    - The Transaction Object referenced by the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection.
- Otherwise:
  - Send XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message upon receiving connection.

3.2.5.3.4 Receiving an XAUSER_XACT_MTAG_ABORT Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_ABORT message, it MUST perform the following actions.

- If the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to Aborted:
  - Send an XAUSER_XACT_MTAG_REQUEST_COMPLETED message.
  - Remove the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection from the XA Superior Enlistment Table of the XA Superior Object indexed in the XA Superior Table by the Resource Manager Recovery GUID of the XA Superior Enlistment Object.
- Otherwise, if the Connection state is set to **Active** and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to either Active, Prepared, or In Doubt:
  - Set the Current Request CMP Connection field of XA Superior Enlistment referenced by the XA Superior Enlistment CMP Connection to receiving connection.
  - Set the State field of XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to Aborting.
  - Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.
    - The Transaction Object referenced by the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection.
- Otherwise:
  - Send XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message on receiving connection.

3.2.5.3.5 Connection Disconnected

When a CONNTYPE_XAUSER_XACT_OPEN connection is disconnected, the XA Subordinate Transaction Manager Facet MUST perform the actions as specified in section 3.1.

3.2.5.3.6 Connection Down

The Connection Down event MUST be signaled with the following argument.

- XA Superior Enlistment CMP Connection

When a CONNTYPE_XAUSER_XACT_OPEN connection is disconnected, the XA Subordinate Transaction Manager Facet SHOULD perform the following actions<18>

- If the State field of the provided XA Superior Enlistment CMP Connection is set to Active and the State field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection is set to Active or Preparing:
  - Set State field of XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection to Aborting.
  - If Current Request CMP Connection field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection is set to the provided XA Superior Enlistment CMP Connection:
    - Reset Current Request CMP Connection field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection.

- Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.
  - The Transaction Object referenced by the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection.

3.2.5.4 CONNTYPE_XAUSER_XACT_MIGRATE as Acceptor

This is an XA Superior Enlistment CMP Connection.

For all messages received in this Connection Type, the XA Subordinate Transaction Manager Facet MUST process the message as specified in section 3.1. The XA Subordinate Transaction Manager Facet MUST also follow the processing rules specified in the following sections.

3.2.5.4.1 Receiving an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE message, it MUST perform the following actions.

- If the connection state is Idle:
  - Attempt to find an XA Superior Object in the XA Superior Table keyed by the **guidXaRm** field of the message.
  - If the Object is not found:
    - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
  - Otherwise:
    - Attempt to find XA Superior Enlistment Object in XA Superior Enlistment Table referenced by the XA Superior Object that meets one of the following conditions:
      - The XID field is set to the **XAIdentifier** field of the message.
      - The Coupling field of the XA Superior Enlistment Object is set to Tight and the XID field of the XA Superior Child Branch Enlistment Object in Child Branch Enlistment Table is set to the value of the **XAIdentifier** field of the message.
    - If an Object is not found:
      - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
    - Otherwise:
      - If first condition was satisfied:
        - If the State field of the XA Superior Enlistment Object is set to Active:
          - Set the State field of XA Superior Enlistment Object to Migrate.
          - Send an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE message.
        - Otherwise:
          - The XA Subordinate Transaction Manager MUST drop the connection.
      - Otherwise, if second condition was satisfied:

- If the State field of the XA Superior Child Branch Enlistment Object is set to Active:
  - Set the State field of XA Superior Child Branch Enlistment Object to Migrate.
  - Send an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE message.
- Otherwise:
  - The XA Subordinate Transaction Manager MUST drop the connection.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.4.2 Receiving an XAUSER_XACT_MTAG_RESUME Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_RESUME message, it MUST perform the following actions.

- If the connection state is Idle:
  - Attempt to find an XA Superior Object in the XA Superior Table referenced by this connection keyed by the **guidXaRm** field of the message.
  - If an Object is not found:
    - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
  - Otherwise:
    - Attempt to find XA Superior Enlistment Object in XA Superior Enlistment Table referenced by the XA Superior Object that meets one of the following conditions.
      - The XID field is set to the **XAIdentifier** field of the message.
      - The Coupling field of the XA Superior Enlistment Object is set to Tight and the XID field of the XA Superior Child Branch Enlistment Object in the Child Branch Table is set to the value of the **XAIdentifier** field of the message.
    - If an Object is not found:
      - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
    - Otherwise:
      - If first condition was satisfied:
        - If the State field of the XA Superior Enlistment Object is set to Migrate:
          - Set State field of the XA Superior Enlistment Object to Active.
          - Send an XAUSER_XACT_MTAG_RESUME_DONE message.
        - Otherwise:
          - Send an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED message.
      - Otherwise, if second condition was satisfied:
        - If the State field of the XA Superior Child Branch Enlistment Object is set to Migrate:

- Set State field of the XA Superior Child Branch Enlistment Object to Active.
- Send an XAUSER_XACT_MTAG_RESUME_DONE message.

- Otherwise:
  - Send an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED message.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.4.3 Connection Disconnected

When a CONNTYPE_XAUSER_XACT_MIGRATE connection is disconnected, the XA Subordinate Transaction Manager Facet MUST perform the actions as specified in section 3.1.

3.2.5.5 CONNTYPE_XAUSER_XACT_BRANCH_START as Acceptor

This is an XA Superior Enlistment CMP Connection.

For all messages received in this Connection Type, the XA Subordinate Transaction Manager Facet MUST process the message as specified in the Common Details section. The XA Subordinate Transaction Manager Facet MUST also follow the processing rules specified in the following sections.

3.2.5.5.1 Receiving an XAUSER_XACT_MTAG_START Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_START message, it MUST perform the following actions.

- If the connection state is Idle:
  - Attempt to find the XA Superior Object in the XA Superior Table keyed by the **guidXaRm** field of the message.
  - If an Object is found:
    - Use this as the XA Superior Object in the remainder of the processing of this event.
  - Otherwise:
    - Attempt to create a new XA Superior Object with the following values.
      - Set the Open Count field to 1.
    - If the creation of the new XA Superior Object fails:
      - Send an XAUSER_XACT_MTAG_START_NO_MEM message using the connection.
    - Otherwise:
      - Add the new XA Superior Object to the XA Superior Table referenced by the XA Subordinate Transaction Manager Facet with the **guidXaRm** field of the message as its key.
      - Attempt to find the XA Superior Enlistment Objects in the Enlistment Table referenced by the Core Transaction Manager Facet that meet the following condition.
        - Resource Manager Recovery GUID set to the **guidXaRm** field of the message.

        - For each XA Superior Enlistment Object found:
            - Add the found XA Superior Enlistment Object to the XA Superior Enlistment Table referenced by the created XA Superior Object.
        - Use the created XA Superior Object as the XA Superior Object in the remainder of the processing of this event.
- Attempt to find the XA Superior Enlistment Object in the XA Superior Enlistment Table referenced by the XA Superior Object that meet both of the following conditions.
    - The XID field is set to the **XAIdentifier** field of the message.
    - The Coupling field is set to Tight.
- If an Object is found:
    - Send an XAUSER_XACT_MTAG_START_DUPLICATE message using the connection.
- Otherwise:
    - Attempt to find the XA Superior Enlistment Object in the XA Superior Enlistment Table referenced by the XA Superior Object that meet all of the following conditions.
        - The GTRID field of the XID field of the XA Superior Enlistment Object is set to the GTRID field of the **XAIdentifier** field of the message.
        - The Coupling field is set to Tight.
        - The State field set to Active or Migrate.
    - If an Object is not found:
        - Attempt to create a new XA Superior Enlistment Object with the following values.
            - The Coupling field is set to Tight.
            - The Resource Manager Recovery GUID is set to **guidXaRm** field of the message.
            - The XID field is set to **XAIdentifier** field of the message.
            - The State field is set to Init.
            - The Current Request CMP connection Object is set to receiving connection.
        - If failed:
            - Send an XAUSER_XACT_MTAG_START_NO_MEM message using the connection.
        - Otherwise:
            - Attempt to create new Transaction with the following settings.
                - The Isolation Level field is set to the **isoLevel** field of the message or 0
                - The Timeout field is set to the **Timeout** field of the message or 0.
                - The Description field is set to the **szDesc** field of the message or "".

- The Isolation Flags field is set to the **isoFlags** field of the message or 0.
- If failed:
  - Send an XAUSER_XACT_MTAG_START_NO_MEM message using the connection.
- Otherwise:
  - Set the Transaction field of the XA Superior Enlistment Object to the newly created Transaction.
  - Signal the Create Superior Enlistment event on the Core Transaction Manager Facet with the following argument.
    - The XA Superior Enlistment Object.

- Otherwise:
  - If the XA Superior Enlistment Object has an XA Superior Child Branch Enlistment Object in Child Branch table with the XID field set to the **XAIdentifier** field of the message:
    - Send an XAUSER_XACT_MTAG_START_DUPLICATE message using the connection.
  - Otherwise:
    - Attempt to create a new XA Superior Child Branch Enlistment Object with the following values.
      - The XID field is set to **XAIdentifier** field of the message.
      - The State field is set to Active.
      - The Parent XA Superior Enlistment Object set to found XA Superior Enlistment Object.
    - If failure:
      - Send an XAUSER_XACT_MTAG_START_NO_MEM message using the connection.
    - Otherwise:
      - Add the XA Superior Child Branch Enlistment Object to the XA Superior Enlistment Object's Child Branch Table.
      - Set the State field of receiving connection to Active.
      - Set the XA Superior Child Branch Enlistment Object reference of the receiving connection to the new XA Superior Child Branch Enlistment.
      - Send an XAUSER_XACT_MTAG_STARTED message with the following arguments.
        - Transaction Identifier of the Transaction Object referenced by the found XA Superior Enlistment Object.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.5.2 Connection Disconnected

When a CONNTYPE_XAUSER_XACT_BRANCH_START connection is disconnected, the XA Subordinate Transaction Manager Facet MUST perform the actions as specified in section 3.1.

3.2.5.5.3 Connection Down

The Connection Down event MUST be signaled with the following argument.

- XA Superior Enlistment CMP Connection

When a CONNTYPE_XAUSER_XACT_BRANCH_START connection is disconnected, the XA Subordinate Transaction Manager Facet SHOULD perform the following actions<19>

- Test if the provided XA Superior Enlistment CMP Connection satisfies the following conditions:
  - State field of the provided XA Superior Enlistment CMP Connection is set to Active.
  - XA Superior Enlistment Object referenced by XA Superior Enlistment CMP Connection references an XA Superior Enlistment Object.
  - State field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection is set to either Active or Preparing.
- If the conditions are met, perform the following actions.
  - Set State field of XA Superior Enlistment Object to Aborting.
  - If Current Request CMP Connection field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection is set to the provided XA Superior Enlistment CMP Connection:
    - Reset Current Request CMP Connection field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection.
  - Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.
    - The Transaction Object referenced by the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection.
- Test if the provided XA Superior Enlistment CMP Connection satisfies the following conditions.
  - State field of the provided XA Superior Enlistment CMP Connection is set to Active
  - XA Superior Enlistment Object referenced by XA Superior Enlistment CMP Connection references an XA Superior Child Branch Enlistment Object
  - State field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection is set to Active
- If the conditions are met perform the following actions.
  - Set State field of Parent XA Superior Enlistment of XA Child Branch Enlistment to Aborting.
  - If Current Request CMP Connection field of XA Child Branch Enlistment Object referenced by provided Connection is set to provided XA Superior Enlistment CMP Connection:

- Reset Current Request CMP Connection field of XA Child Branch Enlistment Object referenced by provided Connection.

- Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.

  - The Transaction Object, referenced by the Parent XA Superior Enlistment Object, referenced by the XA Superior Child Branch Enlistment.

3.2.5.6 CONNTYPE_XAUSER_XACT_BRANCH_OPEN as Acceptor

This is an XA Superior Enlistment CMP Connection.

For all messages received in this Connection Type, the XA Subordinate Transaction Manager Facet MUST process the message as specified in section 3.1. The XA Subordinate Transaction Manager Facet MUST also follow the processing rules specified in the following sections.

3.2.5.6.1 Receiving an XAUSER_XACT_MTAG_OPEN Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_OPEN message, it MUST perform the following actions.

- If the connection state is Idle:

  - Attempt to find an XA Superior Object in the XA Superior Table referenced by this connection keyed by the **guidXaRm** field of the message.

  - If an Object not found:

    - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.

  - Otherwise:

    - Attempt to find XA Superior Enlistment Object in XA Superior Enlistment Table referenced by the XA Superior Object that meets one of the following conditions.

      - The XID field is set to the **XAIdentifier** field of the message.

      - The Coupling field of the XA Superior Enlistment Object is set to Tight.

    - If an Object not found:

      - Attempt to find XA Superior Enlistment Object in XA Superior Enlistment Table referenced by the XA Superior Object that meets one of the following conditions.

        - The GTRID field of the XID field of the XA Superior Enlistment Object is set to the GTRID field of the **XAIdentifier** field of the message.

        - An XA Superior Child Branch Enlistment Object in Child Branch table with the XID field set to the value of the **XAIdentifier** field of the message.

        - The Coupling field of the XA Superior Enlistment Object is set to Tight.

      - If an Object is not found:

        - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.

      - Otherwise:

- Set the State field of receiving connection to Active.
- Set the State field of the XA Superior Child Branch Enlistment Object to Active.
- Set the XA Superior Enlistment Object reference of the receiving connection to the found XA Superior Child Branch Enlistment Object.
- Send an XAUSER_XACT_MTAG_OPENED message with the following argument.
  - Transaction Identifier of the Transaction Object referenced by the provided XA Superior Enlistment Object.

- Otherwise:
  - Set the State field of receiving connection to Active.
  - Set the State field of the XA Superior Enlistment Object to Active.
  - Set the XA Superior Enlistment Object reference of the receiving connection to the found XA Superior Enlistment.
  - Send an XAUSER_XACT_MTAG_OPENED message with the following argument.
    - Transaction Identifier of the Transaction Object referenced by the provided XA Superior Enlistment Object.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.6.2 Receiving an XAUSER_XACT_MTAG_PREPARE Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_PREPARE message, it MUST perform the following actions.

- If the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set:
  - If the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to Active:
    - If the **fSinglePhase** field of the message is set to TRUE:
      - If Child Branch Table referenced by the XA Superior Enlistment Object referenced by the receiving connection:
        - Send XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message on receiving connection.
        - The processing for this message is complete.
      - Set the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to Preparing Single Phase.
    - Otherwise:
      - Set State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to Preparing.

- Set Current Request CMP Connection field of the XA Superior Enlistment referenced by the XA Superior Enlistment CMP Connection to receiving connection.
- If Wait For All XA Branch Prepares flag is not set:
  - Signal the Begin Phase Zero event on the Core Transaction Manager Facet with the following argument.
    - The Transaction Object, referenced by the XA Superior Enlistment Object, referenced by the XA Superior Enlistment CMP Connection.

- Otherwise, if the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to Aborted:
  - Send XAUSER_XACT_MTAG_PREPARE_ABORT message.
  - Remove the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection from the XA Superior Enlistment Table of the XA Superior Object indexed in the XA Superior Table by the Resource Manager Recovery GUID of the XA Superior Enlistment Object.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

- Otherwise, if the XA Superior Child Branch Enlistment Object referenced by the receiving connection is set:
  - If the **fSinglePhase** field of the message is set to TRUE:
    - Send XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message on receiving connection.
    - The processing for this message is complete.
  - If the Connection state is set to Active and the State field of the XA Superior Child Branch Enlistment Object is set to Active:
    - Remove the XA Superior Child Branch Enlistment from the Child Branch Table referenced by its Parent XA Superior Enlistment Object.
    - Send XAUSER_XACT_MTAG_READONLY message on receiving connection.
    - If the Child Branch Table referenced by the Parent XA Superior Enlistment Object referenced by the XA Superior Child Branch Enlistment Object is empty and Wait For All XA Branch Prepares flag is set:
      - Signal the Begin Phase Zero event on the Core Transaction Manager Facet with the following argument.
        - The Transaction Object referenced by the Parent XA Superior Enlistment Object referenced by the XA Superior Child Branch Enlistment Object.
- Otherwise:
  - Send XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message upon receiving connection.

3.2.5.6.3 Receiving an XAUSER_XACT_MTAG_COMMIT Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_COMMIT message, it MUST perform the following actions.

- If the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to either Prepared or In Doubt:
  - Set Current Request CMP Connection field of XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to receiving connection.
  - Set State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to Committing.
  - Signal the Begin Commit event on the Core Transaction Manager Facet with the following argument.
    - The Transaction Object, referenced by the XA Superior Enlistment Object, referenced by the XA Superior Enlistment CMP Connection.
- Otherwise:
  - Send a XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message upon receiving connection.

3.2.5.6.4 Receiving an XAUSER_XACT_MTAG_ABORT Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_ABORT message, it MUST perform the following actions.

- If the receiving connection references XA Superior Enlistment:
  - If the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to Aborted:
    - Send an XAUSER_XACT_MTAG_REQUEST_COMPLETED message.
    - Remove XA Superior Enlistment Object from the XA Superior Enlistment Table of the XA Superior Object indexed in the XA Superior Table by the Resource Manager Recovery GUID of the XA Superior Enlistment Object.
  - Otherwise if the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to either Active, Prepared, or In Doubt:
    - Set Current Request CMP Connection field of XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to receiving connection.
    - Set State field of XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection to Aborting.
    - Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.
      - The Transaction Object referenced by the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection.

- Otherwise, if the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by the XA Superior Enlistment CMP Connection is set to Aborting and Current Request CMP Connection referenced by XA Superior Enlistment referenced by the XA Superior Enlistment CMP Connection is not set:
  - Set Current Request CMP Connection field of XA Superior Enlistment referenced by the XA Superior Enlistment CMP Connection to receiving connection.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

- Otherwise, if receiving connection references XA Superior Child Branch Enlistment:
  - If the Connection state is set to Active and the State field of the Parent XA Superior Enlistment Object referenced by XA Superior Child Branch Enlistment referenced by receiving connection is set to Aborted:
    - Send XAUSER_XACT_MTAG_REQUEST_COMPLETED message.
    - Remove XA Superior Child Branch Enlistment from XA Superior Child Branch Enlistment Table referenced by Parent XA Superior Enlistment Object referenced by XA Superior Child Branch Enlistment.
  - Otherwise, if the Connection state is set to Active and the State field of the XA Superior Enlistment is set to either Active, Prepared, or In Doubt:
    - Set Current Request CMP Connection field of the XA Superior Child Branch Enlistment to receiving connection.
    - Set the State field of Parent XA Superior Enlistment referenced by XA Superior Child Branch Enlistment to Aborting.
    - Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.
      - The Transaction Object referenced by Parent XA Superior Enlistment Object.
  - Otherwise, if the Connection state is set to Active and the State field of the XA Superior Enlistment Object is set to Aborting and Current Request CMP Connection referenced by XA Superior Enlistment is not set:
    - Set Current Request CMP Connection field of XA Superior Child Branch Enlistment to receiving connection.
  - Otherwise:
    - Send a XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message upon receiving connection.

3.2.5.6.5 Connection Disconnected

When a CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection is disconnected, the XA Subordinate Transaction Manager Facet MUST perform the actions as specified in section 3.1.5.

3.2.5.6.6 Connection Down

The Connection Down event MUST be signaled with the following argument.

- XA Superior Enlistment CMP Connection

When a CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection is disconnected, the XA Subordinate Transaction Manager Facet SHOULD perform the following actions<20>

- If the Connection state is set to Active and the State field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection is set to either Active or Preparing:
  - Set the State field of XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection to Aborting.
  - If Current Request CMP Connection field of the XA Superior Enlistment referenced by provided XA Superior Enlistment CMP Connection is set to provided XA Superior Enlistment CMP Connection:
    - Reset Current Request CMP Connection field of the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection.
  - Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.
    - The Transaction Object referenced by the XA Superior Enlistment Object referenced by provided XA Superior Enlistment CMP Connection.
- If the Connection state is set to Active and the State field of the XA Superior Child Branch Enlistment Object is set to either Active or Preparing.
  - Set State field of Parent XA Superior Enlistment Object of XA Superior Child Branch Enlistment to Aborting.
  - If Current Request CMP Connection field of the XA Child Branch Enlistment Object referenced by provided Connection is set to provided XA Superior Enlistment CMP Connection:
    - Reset Current Request CMP Connection field of XA Child Branch Enlistment Object referenced by the provided Connection.
  - Signal the Begin Rollback event on the Core Transaction Manager Facet with the following argument.
    - The Transaction Object referenced by the Parent XA Superior Enlistment of the XA Child Branch Enlistment.

3.2.5.7 CONNTYPE_XAUSER_XACT_MIGRATE2 as Acceptor

This is an XA Superior Enlistment CMP Connection.

For all messages received in this Connection Type, the XA Subordinate Transaction Manager Facet MUST process the message as specified in section 3.1. The XA Subordinate Transaction Manager Facet MUST also follow the processing rules specified in the following sections.

3.2.5.7.1 Receiving an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE message, it MUST perform the following actions.

- If the connection state is Idle:

- Attempt to find an XA Superior Object in the XA Superior Table keyed by the **guidXaRm** field of the message.
- If an Object is not found:
  - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
- Otherwise:
  - Attempt to find an XA Superior Enlistment Object in the XA Superior Enlistment Table referenced by the XA Superior Object that meets one of the following conditions.
    - The XID field is set to the value of the **XAIdentifier** field in the message.
    - The Coupling field of the XA Superior Enlistment Object is set to Tight and the XA Superior Child Branch Enlistment Object in Child Branch Enlistment Table has the XID field set to the **XAIdentifier** field of the message.
  - If an Object is not found:
    - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
  - Otherwise:
    - If first condition was satisfied:
      - If the State field of the XA Superior Enlistment Object is set to Active:
        - Set the State field of the XA Superior Enlistment Object to Migrate.
        - Send an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE message.
      - Otherwise:
        - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message.
    - Otherwise if second condition was satisfied:
      - If the State field of the XA Superior Child Branch Enlistment is set to Active:
        - Set the State field of the XA Superior Child Branch Enlistment to Migrate.
        - Send an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE message.
      - Otherwise:
        - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.7.2 Receiving an XAUSER_XACT_MTAG_RESUME Message

When the XA Subordinate Transaction Manager Facet receives an XAUSER_XACT_MTAG_RESUME message, it MUST perform the following actions.

- If the connection state is Idle:
  - Attempt to find an XA Superior Object in the XA Superior Table keyed by the **guidXaRm** field of the message.

- If an Object is not found:
  - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
- Otherwise:
  - Attempt to find XA Superior Enlistment Object in XA Superior Enlistment Table referenced by the XA Superior Object that meets one of the following conditions.
    - The XID field is set to the value of the **XAIdentifier** field of the message.
    - The Coupling field of the XA Superior Enlistment Object is set to Tight and XA Superior Child Branch Enlistment Object in the Child Branch Enlistment Table with the XID value same as the **XAIdentifier** field of the message
  - If an Object is not found:
    - Send an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message using the connection.
  - Otherwise:
    - If first condition was satisfied:
      - If State field of the XA Superior Enlistment Object is set to Migrate:
        - Set the State field of the XA Superior Enlistment Object to Active.
        - Send an XAUSER_XACT_MTAG_RESUME_DONE message with the following value.
          - Transaction Identifier of the Transaction Object referenced by the found XA Superior Enlistment Object.
      - Otherwise:
        - Send an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED message.
    - Otherwise if second condition was satisfied:
      - If State field of the XA Superior Child Branch Enlistment Object is set to Migrate:
        - Set the State field of the XA Superior Child Branch Enlistment to Active.
        - Send an XAUSER_XACT_MTAG_RESUME_DONE message with the following argument.
          - Transaction Identifier of the Transaction Object referenced by the found XA Superior Enlistment Object.
      - Otherwise:
        - Send an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED message.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.2.5.7.3 Connection Disconnected

When a CONNTYPE_XAUSER_XACT_MIGRATE2 connection is disconnected, the XA Subordinate Transaction Manager Facet MUST perform the actions as specified in section 3.1.

3.2.6 Timer Events

This role has no protocol-specific timer events.

3.2.7 Other Local Events

3.2.7.1 Commit Complete

The Commit Complete event MUST be signaled with the following arguments.

- An XA Superior Enlistment Object.

If the Commit Complete event is signaled, the XA Subordinate Transaction Manager Facet MUST perform the following actions.

- If the State field of the provided XA Superior Enlistment Object is set to Committing:
  - Set the State field of the provided XA Superior Enlistment Object to Committed.
  - Send an XAUSER_XACT_MTAG_REQUEST_COMPLETED message.
  - Reset the Current Request CMP Connection field of the provided XA Superior Enlistment Object.
  - Remove the provided XA Superior Enlistment Object from the XA Superior Enlistment Table, referenced by the XA Superior Object indexed in the XA Superior Table by the Resource Manager Recovery GUID field of the provided XA Superior Enlistment Object.

3.2.7.2 Create Superior Enlistment Success

The Create Superior Enlistment Success event MUST be signaled with the following arguments.

- An XA Superior Enlistment Object.

If the Create Superior Enlistment Success event is signaled, the XA Subordinate Transaction Manager Facet MUST perform the following actions.

- If the State field of the provided XA Superior Enlistment Object is set to Init:
  - Attempt to find XA Superior Object in XA Superior Table keyed by the Resource Manager Recovery GUID field of the provided XA Superior Enlistment Object.
  - If an Object is not found:
    - Attempt to create a new XA Superior Object with the following values.
      - The Open Count field is set to 1.
      - The Current Recovery XID is set to the Resource Manager Recovery GUID field of the provided XA Superior Enlistment Object.
    - If creation of the new XA Superior Object fails:
      - Send an XAUSER_XACT_MTAG_START_NO_MEM message using the connection.
      - The processing for this event is complete.

- Add the provided XA Superior Enlistment Object to the XA Superior Enlistment Table referenced by the XA Superior Object indexed in the XA Superior Table by the Resource Manager Recovery GUID field of XA Superior Enlistment Object.
- Set the State field of receiving connection to Active.
- Set the State field of the provided XA Superior Enlistment Object to Active.
- Set the XA Superior Enlistment Object reference of the XA Superior Enlistment's Current Request CMP Connection to the located XA Superior Enlistment Object.
- Reset XA Superior Enlistment's Current Request CMP Connection Object.
- Send an XAUSER_XACT_MTAG_STARTED message with the following arguments.
  - The Transaction Identifier of the Transaction Object referenced by the provided XA Superior Enlistment Object.

- Otherwise, the event MUST be processed as invalid as specified in section 3.1.5.

3.2.7.3 Create Superior Enlistment Failure

The Create Superior Enlistment Failure event MUST be signaled with the following arguments.

- An XA Superior Enlistment Object.
- Failure Reason.

If the Create Superior Enlistment Failure event is signaled, the XA Subordinate Transaction Manager Facet MUST perform the following actions.

- If the State field of the provided XA Superior Enlistment Object is set to Init:
  - If Failure Reason is set to INITXASUPERIOR_LOG_FULL:
    - Send an XAUSER_XACT_MTAG_START_LOG_FULL message
  - Otherwise if Failure Reason is set to INITXASUPERIOR_NO_MEM:
    - Send an XAUSER_XACT_MTAG_START_NO_MEM message
  - Delete Transaction and the provided XA Superior Enlistment Objects.
- Otherwise, the event MUST be processed as invalid as specified in the Message Processing Events and Sequencing Rules section.

3.2.7.4 Phase Zero Complete

The Phase Zero Complete event MUST be signaled with the following arguments.

- An XA Superior Enlistment Object
- Outcome

If the Phase Zero Complete event is signaled, the XA Subordinate Transaction Manager Facet MUST perform the following actions.

- If the provided Outcome is set to Failure and the Current Request CMP Connection Object referenced by the provided XA Superior Enlistment Object is set:
  - Send an XAUSER_XACT_MTAG_PREPARE_ABORT message.
  - Set State of provided XA Superior Enlistment Object to ABORTED.
  - Reset Current Request CMP Connection of the provided XA Superior Enlistment Object.
  - Remove the provided XA Superior Enlistment Object from the XA Superior Enlistment Table referenced by the XA Superior Object indexed in the XA Superior Table by the Resource Manager Recovery GUID field of XA Superior Enlistment Object.
- Otherwise:
  - If the State field of the provided XA Superior Enlistment Object is set to Preparing Single Phase:
    - Signal the Begin Phase One event on the Core Transaction Manager Facet with the following arguments.
      - The provided XA Superior Enlistment Object.
      - A Single Phase value set to TRUE.
  - Otherwise if the State field of the provided XA Superior Enlistment Object is set to Preparing:
    - Signal the Begin Phase One event on the Core Transaction Manager Facet with the following arguments.
      - The Transaction Object referenced by the provided XA Superior Enlistment Object.
      - A Single Phase value set to FALSE.

3.2.7.5 Phase One Complete

The Phase One Complete event MUST be signaled with the following arguments.

- An XA Superior Enlistment Object
- Outcome

If the Phase One Complete event is signaled, the XA Subordinate Transaction Manager Facet MUST perform the following actions.

- If the Current Request CMP Connection Object referenced by the provided XA Superior Enlistment Object is set:
  - If the provided Outcome is set to **In Doubt**:
    - Set the State field of the provided XA Superior Enlistment Object to In Doubt
    - Send an XAUSER_XACT_MTAG_PREPARE_SINGLEPHASE_INDOUBT message.
    - Reset the Current Request CMP Connection Object referenced by the provided XA Superior Enlistment Object.

- Otherwise if the provided Outcome value is set to Committed and the State field of the provided XA Superior Enlistment Object is set to Preparing Single Phase:
  - Set the State field of the provided XA Superior Enlistment Object to Committed.
  - Send an XAUSER_XACT_MTAG_REQUEST_COMPLETED message.
  - Reset the Current Request CMP Connection Object referenced by the provided XA Superior Enlistment Object.
  - Remove the provided XA Superior Enlistment Object from the XA Superior Enlistment Table referenced by the XA Superior Object indexed in the XA Superior Table by the Resource Manager Recovery GUID field of XA Superior Enlistment Object.
- Otherwise if Outcome value is set to Aborted:
  - Set the State field of the provided XA Superior Enlistment Object to Aborted.
  - Send an XAUSER_XACT_MTAG_PREPARE_ABORT message.
  - Reset the Current Request CMP Connection Object referenced by the provided XA Superior Enlistment Object.
  - Remove the provided XA Superior Enlistment from XA Superior's XA Superior Enlistment Table.
- Otherwise if the Outcome value is set to Prepared:
  - Set the State field of the provided XA Superior Enlistment Object to Prepared.
  - Send an XAUSER_XACT_MTAG_REQUEST_COMPLETED message.
  - Reset the Current Request CMP Connection Object referenced by the provided XA Superior Enlistment Object.
- Otherwise if Outcome value is set to ReadOnly:
  - The XA Subordinate Transaction Manager Facet will drop the connection.

3.2.7.6 Recover In Doubt Transaction

If the Recover In Doubt event is signaled, the XA Subordinate Transaction Manager Facet MUST take no action.

3.2.7.7 Rollback Complete

The Rollback Complete event MUST be signaled with the following arguments.

- An XA Superior Enlistment Object.

If the Rollback Complete event is signaled, the XA Subordinate Transaction Manager Facet MUST perform the following actions.

- If the State field of the provided XA Superior Enlistment Object is set to Aborting:
  - Set the State field of the provided XA Superior Enlistment Object to Aborted.

- If Current Request CMP Connection Object of the provided XA Superior Enlistment Object is set:
  - Send XAUSER_XACT_MTAG_REQUEST_COMPLETED message.
  - Reset Current Request CMP Connection of the provided XA Superior Enlistment Object.
  - Remove the provided XA Superior Enlistment Object from the XA Superior Enlistment Table referenced by the XA Superior Object indexed in the XA Superior Table by the Resource Manager Recovery GUID field of XA Superior Enlistment Object.
- Perform the following actions on each XA Superior Child Branch Enlistment Object in the XA Superior Child Branch Enlistment Table referenced by XA Superior Enlistment Object:
  - If Current Request CMP Connection Object of XA Superior Child Branch Enlistment Object is set:
    - Send XAUSER_XACT_MTAG_REQUEST_COMPLETED message.
    - Reset Current Request CMP Connection of XA Superior Child Branch Enlistment Object.
    - Remove XA Superior Enlistment from XA Superior Child Branch Enlistment Table referenced by Parent XA Superior Enlistment Object.
- Otherwise, the event MUST be processed as invalid as specified in section 3.1.5.

3.2.7.8 Unilaterally Aborted

The Unilaterally Aborted event MUST be signaled with the following arguments.

- An XA Superior Enlistment Object.

If the Unilaterally Aborted event is signaled, the XA Subordinate Transaction Manager Facet MUST perform the following actions.

- If the State field of the provided XA Superior Enlistment Object is set to Active:
  - Set the State field of the provided XA Superior Enlistment Object to Aborted.
- Otherwise, the event MUST be processed as invalid as specified in section 3.1.5.

3.3 XA Superior Transaction Manager

3.3.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate the implementations adhere to this model as long as their external behavior is consistent with the behavior that is described in this document.

Note that the abstract data model can be implemented in a variety of ways. This protocol does not prescribe or advocate any specific implementation technique.

The XA Superior Transaction Manager MUST maintain all the data elements that are specified in section 3.1.1.

The Superior Transaction Manager MUST also maintain the following data elements.

- **XA Superior Proxy object**: This object represents a Superior XA Transaction Manager associated with a Resource Manager ID. An XA Superior Proxy object MUST contain the following elements.
  - **Transaction Manager Connection:** A connection object as defined by [MS-CMP] to hold the connection with the XA Subordinate Transaction Manager Facet
  - **Open Count**: Specifies how many times xa_open has been called in association with the current Resource Manager ID.
  - Resource Manager Recovery GUID: An identifier provided to identify the XA Superior Transaction Manager to the XA Subordinate Transaction Manager Facet. This identifier persists through failure and recovery.
  - **Branch Isolation**: A flag specifying whether the associated XA Branches corresponding to the XA Transaction objects in the XA Transaction table are loosely or tightly coupled.
    - **Tight:** Transaction objects are associated to XIDs based on the GTRID field of the XID.
    - **Loose:** Transaction objects are associated to XIDs based on the entire XID.
  - TM: A transaction manager description, used for the Transaction object.
  - **Recovery Complete {true,false}**: A Boolean flag indicating if the current recovery process is complete. This flag can have one of the two values: TRUE or FALSE. If there was no recovery started or if the recovery process has ended its value will be TRUE otherwise FALSE.
  - **XA Transaction Table**: A table of XA Transaction objects associated with the resource manager ID keyed by XID.
  - **Recovery Request Number**: The number of XIDs requested from XA Subordinate Transaction Manager Facet in the most recent XAUSER_CONTROL_MTAG_RECOVER message.
  - **Recovery Max Return**: The maximum number of XIDs to return to xa_recover event triggered on the XA Superior Transaction Manager.
  - **Recovery Return Number**: The number of XIDs in need of Recovery that have been received from the XA Subordinate Transaction Manager Facet.
  - **Recovery Array**: A reference to an array of XIDs in need of Recovery that have been received from the XA Subordinate Transaction Manager Facet.
- **XA Superior Proxy Table:** A table of XA Superior Proxy objects that is keyed by Resource Manager ID.
- **XA Transaction object:** This object represents an XA Transactions Branch. An XA Transactions object MUST contain the following elements.
  - XID: The XID associated with the XA Transaction
  - Transaction Identifier: The identifier field of the OleTx transaction object associated with the XA Transaction
  - **Thread Identifier: The** execution Thread Identifier used to ensure thread affinity if required by setting Require Thread Affinity attribute to TRUE.

- **Migrate:** A Boolean flag indicating if the XA Transaction has been migrated. This flag can take one of the two values: TRUE or FALSE.
- **Require Thread Affinity:** A Boolean flag indicating if additional validation is to be performed to enforce thread affinity for the XA Transaction. This flag can take one of the two values: TRUE or FALSE.
- **Parent XA Superior Proxy Object:** A reference to the corresponding XA Superior Proxy object.
- **State:** The current state of the XA Transaction. This attribute can take of the following values. Idle, Starting, Opening, Preparing, Prepared, Committing, Aborting, Active or Suspended.
- **Connection:** A reference to XA Transaction CMP Connection object.

- **XA Superior Proxy CMP Connection object:** CMP connection object extended to include the following:
  - A reference to an XA Superior Proxy object
- **XA Transaction CMP Connection object:** CMP Connection object extended to include the following:
  - Reference to an XA Transaction Object
  - Action**:**
    - {Prepare, Prepare Single Phase, Commit, Rollback, None, Migrate, Resume}

3.3.1.1 Versioning

The XA Superior Transaction Manager MUST maintain the data that pertains to the migrate functionality only on versions where the connection type CONNTYPE_XAUSER_XACT_MIGRATE or CONNTYPE_XAUSER_XACT_MIGRATE2 is supported as specified in 2.2.2. The following data elements, as specified in 3.2.1, are affected.

- XA Transaction Object
  - Migrate
- XA Transaction CMP Connection Object
  - Action
    - Migrate

3.3.1.2 CONNTYPE_XA_USER_CONTROL Initiator States

The XA Superior Transaction Manager MUST act as an initiator for the CONNTYPE_XA_USER_CONTROL connection type. In this role, the XA Superior Transaction Manager MUST provide support for the following states.

- Idle
- Awaiting Creation Response
- Active

- Awaiting Recovery Response
- Ended

3.3.1.2.1 Idle

This is the initial state.

3.3.1.2.2 Awaiting Creation Response

The following events are processed in this state.

- Receiving an _XAUSER_CONTROL_MTAG_CREATED Message
- Receiving an _XAUSER_CONTROL_MTAG_CREATE_NO_MEM Message.
- Connection Down
- Connection Disconnected

3.3.1.2.3 Active

The following event is processed in this state.

- Connection Down

3.3.1.2.4 Awaiting Recovery Response

The following events are processed in this state.

- Receiving an _XAUSER_CONTROL_MTAG_CREATE_NO_MEM Message
- Receiving an _XAUSER_CONTROL_MTAG_RECOVER_REPLY Message
- Connection Down
- Connection Disconnected

3.3.1.2.5 Ended

This is the final state.

3.3.1.2.6 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_CONTROL initiator states.

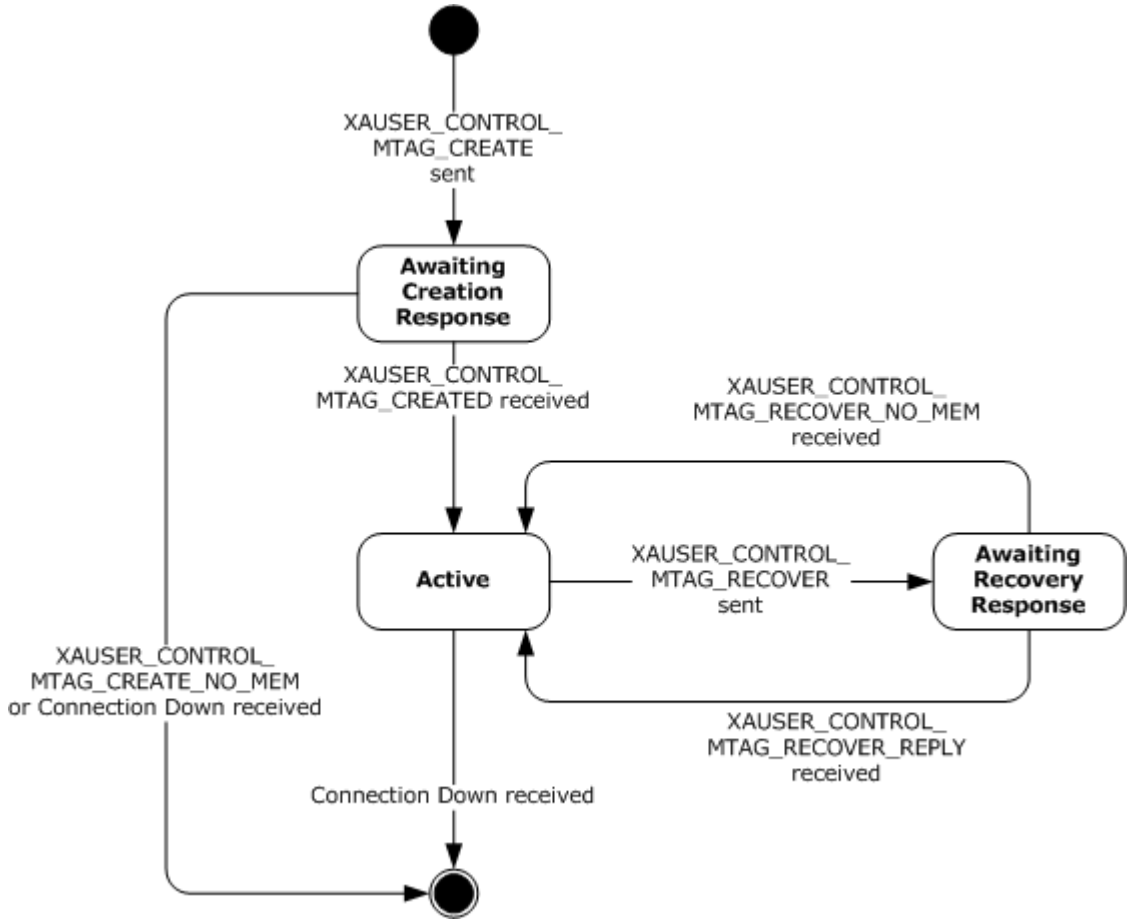


Figure 19: Relationship between the CONNTYPE_ XAUSER_XACT_CONTROL initiator states

3.3.1.3 CONNTYPE_XAUSER_XACT_START Initiator States

The XA Superior Transaction Manager MUST act as an initiator for the CONNTYPE_XA_USER_XACT_START connection type. In this role, the XA Superior Transaction Manager MUST provide support for the following states.

- Idle
- Awaiting Start Response
- Active
- Ended

3.3.1.3.1 Idle

This is the initial state.

3.3.1.3.2 Awaiting Start Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_STARTED Message
- Receiving an XAUSER_XACT_MTAG_START_NO_MEM Message
- Receiving an XAUSER_XACT_MTAG_START_LOG_FULL Message
- Receiving an_XAUSER_XACT_MTAG_START_DUPLICATE Message
- Connection Disconnected
- Connection Down

3.3.1.3.3 Active

The following events are processed in this state.

- Connection Disconnected.
- Connection Down.

3.3.1.3.4 Ended

This is the final state.

3.3.1.3.5 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_START initiator states.

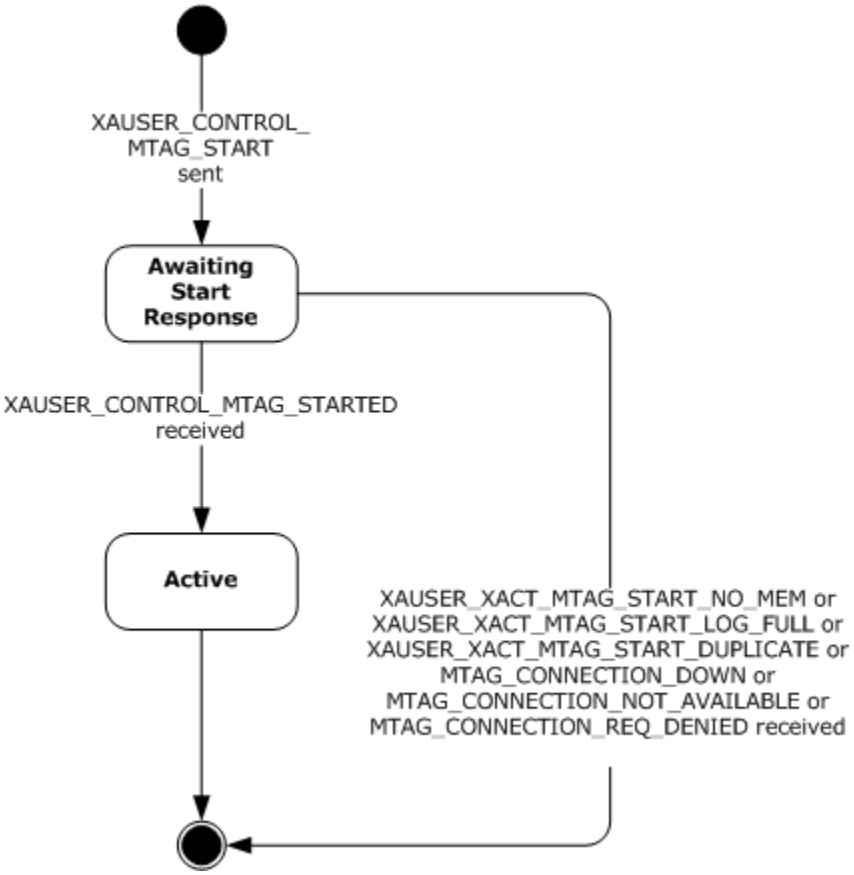


Figure 20: CONNTYPE_ XAUSER_XACT_START initiator states

3.3.1.4 CONNTYPE_XAUSER_XACT_OPEN Initiator States

The XA Superior Transaction Manager MUST act as an initiator for the CONNTYPE_XA_USER_XACT_OPEN connection type. In this role, the XA Superior Transaction Manager MUST provide support for the following states.

- Idle.
- Awaiting Open Response.
- Active.
- Awaiting Prepare Response.
- Awaiting Abort Response.
- Awaiting Commit Response.
- Ended.

3.3.1.4.1 Idle

This is the initial state.

3.3.1.4.2 Awaiting Open Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_OPENED Message
- Receiving an XAUSER_XACT_MTAG_OPEN_NOT_FOUND Message
- Connection Disconnected
- Connection Down

3.3.1.4.3 Active

The following events are processed in this state,

- Connection Disconnected.
- Connection Down.

3.3.1.4.4 Awaiting Prepare Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_REQUEST_COMPLETED Message
- Receiving an XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL Message
- Receiving an XAUSER_XACT_MTAG_PREPARE_ABORT Message
- Receiving an XAUSER_XACT_MTAG_PREPARE_SINGLEPHASE_INDOUBT
- Connection Disconnected

3.3.1.4.5 Awaiting Abort Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_REQUEST_COMPLETED Message
- Receiving an XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL Message
- Connection Disconnected
- Connection Down

3.3.1.4.6 Awaiting Commit Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_REQUEST_COMPLETED Message
- Receiving an XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL Message
- Connection Disconnected
- Connection Down

3.3.1.4.7 Ended

This is the final state.

3.3.1.4.8 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_OPEN initiator states.

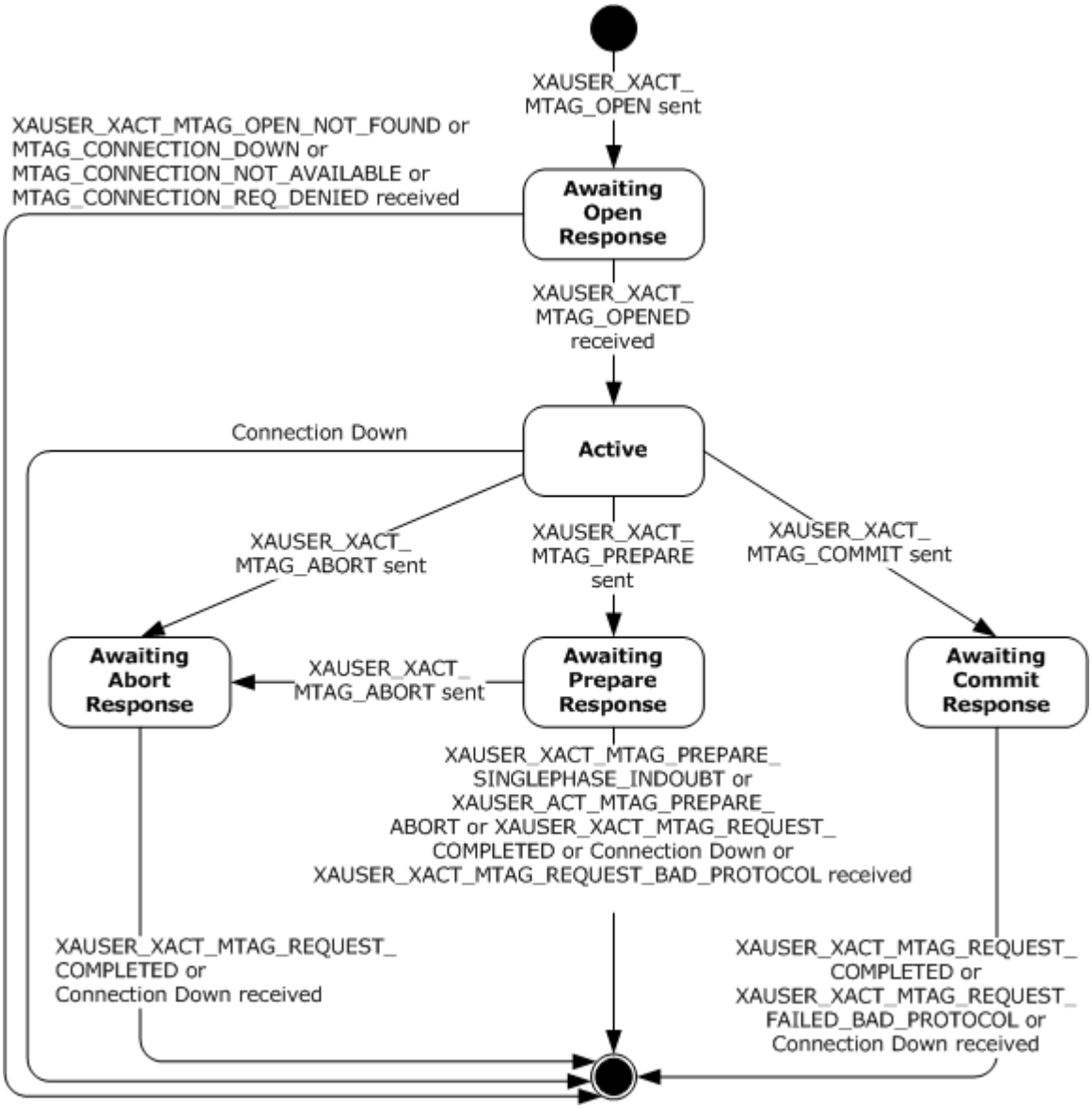


Figure 21: CONNTYPE_ XAUSER_XACT_OPEN initiator states

3.3.1.5 CONNTYPE_XAUSER_XACT_MIGRATE Initiator States

The XA Superior Transaction Manager MUST act as an initiator for the CONNTYPE_XA_USER_XACT_MIGRATE connection type. In this role, the XA Superior Transaction Manager MUST provide support for the following states.

- Idle.
- Awaiting Suspension Response.
- Awaiting Resumption Response.
- Ended.

3.3.1.5.1 Idle

This is the initial state.

3.3.1.5.2 Awaiting Suspension Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE Message
- Receiving an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED Message
- Connection Disconnected
- Connection Down

3.3.1.5.3 Awaiting Resumption Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_RESUME_DONE Message
- Receiving an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED Message
- Connection Disconnected
- Connection Down

3.3.1.5.4 Ended

This is the final state.

3.3.1.5.5 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_MIGRATE initiator states.

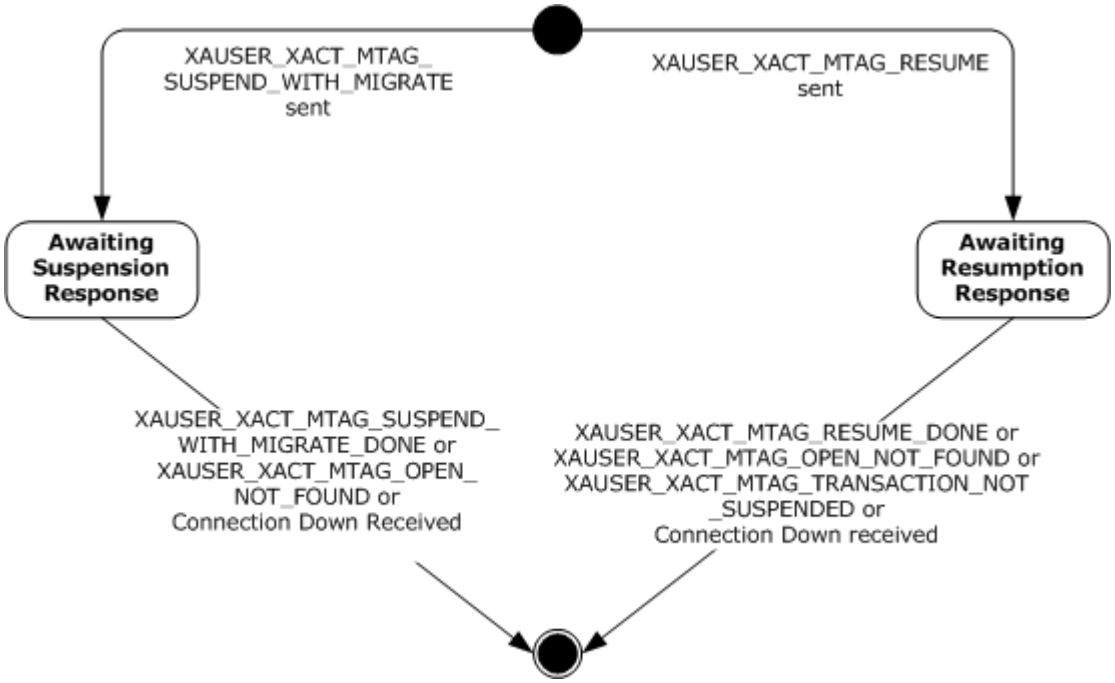


Figure 22: CONNTYPE_ XAUSER_XACT_MIGRATE initiator states

3.3.1.6 CONNTYPE_XAUSER_XACT_BRANCH_START Initiator States

The XA Superior Transaction Manager MUST act as an initiator for the CONNTYPE_XA_USER_XACT_BRANCH_START connection type. In this role, the XA Superior Transaction Manager MUST provide support for the following state.

- Idle.
- Awaiting Start Response.
- Active.
- Ended.

3.3.1.6.1 Idle

This is the initial state.

3.3.1.6.2 Awaiting Start Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_STARTED_Message
- Receiving an XAUSER_XACT_MTAG_START_NO_MEM Message
- Receiving an XAUSER_XACT_MTAG_START_LOG_FULL_Message
- Receiving an XAUSER_XACT_MTAG_START_DUPLICATE Message

- Connection Disconnected
- Connection Down

3.3.1.6.3 Active

The following events are processed in this state.

- Connection Disconnected
- Connection Down

3.3.1.6.4 Ended

This is the final state.

3.3.1.6.5 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_BRANCH_START initiator states.

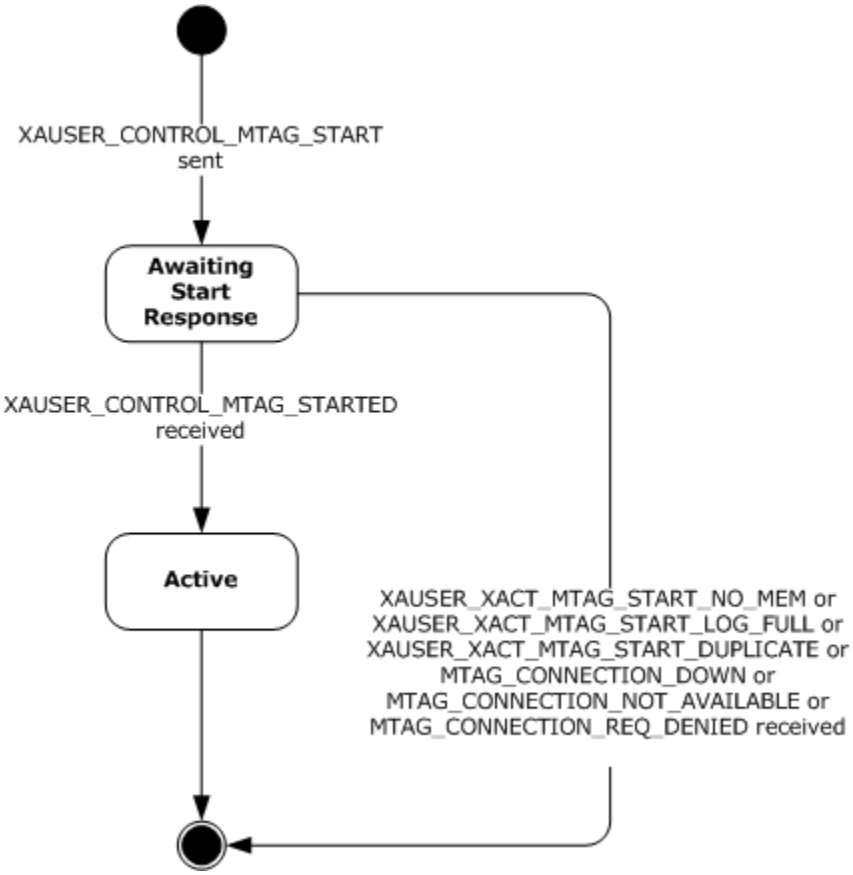


Figure 23: CONNTYPE_ XAUSER_XACT_BRANCH_START initiator states

3.3.1.7 CONNTYPE_XAUSER_XACT_BRANCH_OPEN Initiator States

The XA Superior Transaction Manager MUST act as an initiator for the CONNTYPE_XA_USER_XACT_BRANCH_OPEN connection type. In this role, the XA Superior Transaction Manager MUST provide support for the following states.

- Idle.
- Awaiting Open Response.
- Active.
- Awaiting Prepare Response.
- Awaiting Abort Response.
- Awaiting Commit Response.
- Ended

3.3.1.7.1 Idle

This is the initial state.

3.3.1.7.2 Awaiting Open Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_OPENED Message
- Receiving an XAUSER_XACT_MTAG_OPEN_NOT_FOUND Message
- Connection Disconnected
- Connection Down

3.3.1.7.3 Active

The following events are processed in this state.

- Connection Disconnected
- Connection Down

3.3.1.7.4 Awaiting Prepare Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_REQUEST_COMPLETED Message
- Receiving an XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL Message
- Receiving an XAUSER_XACT_MTAG_PREPARE_ABORT Message
- Receiving an XAUSER_XACT_MTAG_PREPARE_SINGLEPHASE_INDOUBT Message
- Receiving an XAUSER_XACT_MTAG_READONLY_ Message

- Connection Disconnected
- Connection Down

3.3.1.7.5 Awaiting Abort Response

The following events are processed in this state.

- Receiving an _XAUSER_XACT_MTAG_REQUEST_COMPLETED Message
- Connection Disconnected
- Connection Down

3.3.1.7.6 Awaiting Commit Response

The following events are processed in this state.

- Receiving an _XAUSER_XACT_MTAG_REQUEST_COMPLETED Message
- Connection Disconnected
- Connection Down

3.3.1.7.7 Ended

This is the final state.

3.3.1.7.8 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_BRANCH_OPEN initiator states.

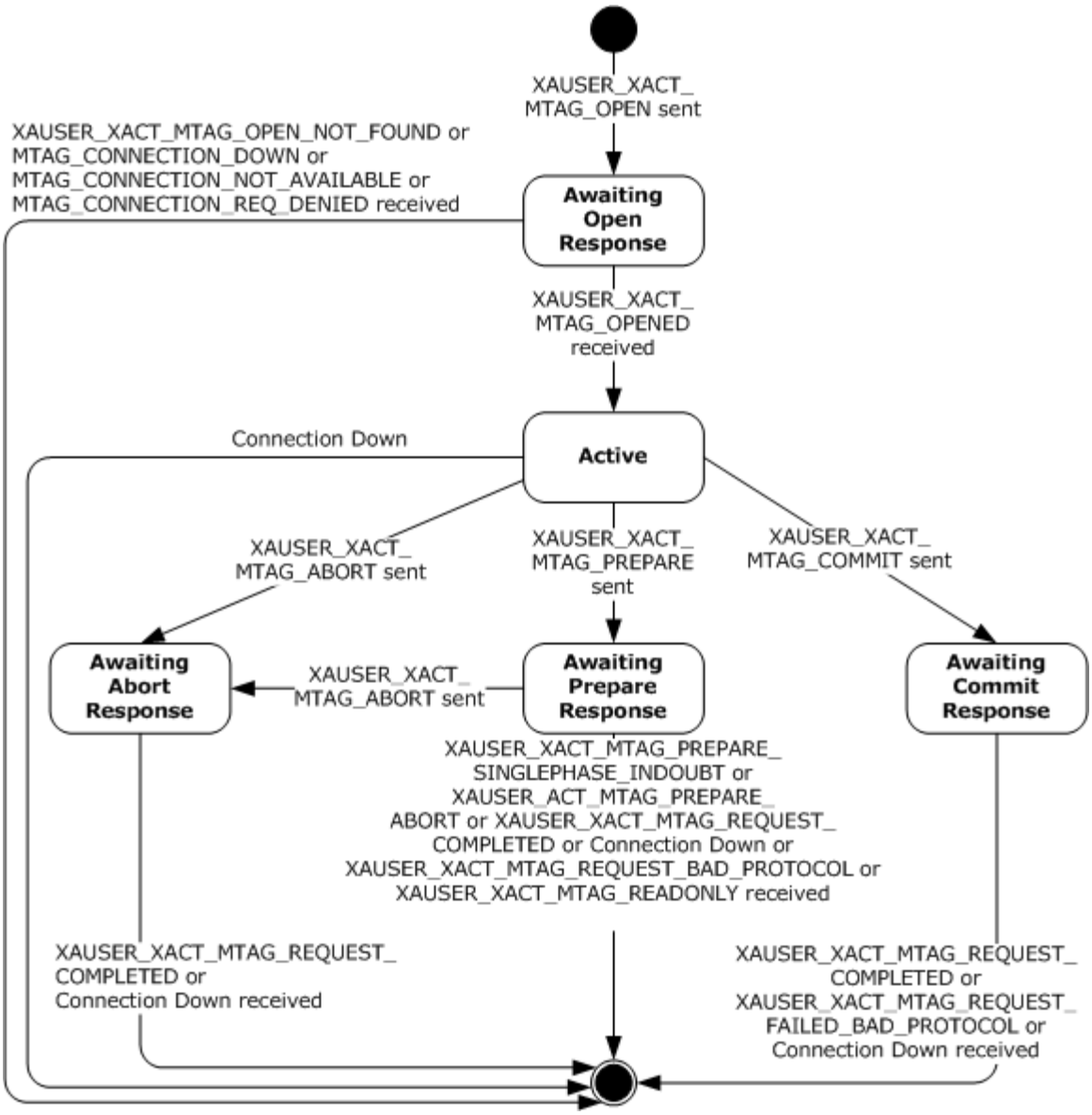


Figure 24: CONNTYPE_ XAUSER_XACT_BRANCH_OPEN initiator states

3.3.1.8 CONNTYPE_XAUSER_XACT_MIGRATE2 Initiator States

The XA Superior Transaction Manager MUST act as an initiator for the CONNTYPE_XA_USER_XACT_MIGRATE2 connection type. In this role, the XA Superior Transaction Manager MUST provide support for the following states.

- Idle
- Awaiting Suspension Response.
- Awaiting Resumption Response.

- Ended.

3.3.1.8.1 Idle

This is the initial state.

3.3.1.8.2 Awaiting Suspension Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE Message
- Connection Disconnected
- Connection Down

3.3.1.8.3 Awaiting Resumption Response

The following events are processed in this state.

- Receiving an XAUSER_XACT_MTAG_RESUME_DONE Message
- Receiving an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED Message
- Connection Disconnected
- Connection Down

3.3.1.8.4 Ended

This is the final state.

3.3.1.8.5 State Diagram

The following figure shows the relationship between the CONNTYPE_ XAUSER_XACT_MIGRATE2 initiator states.

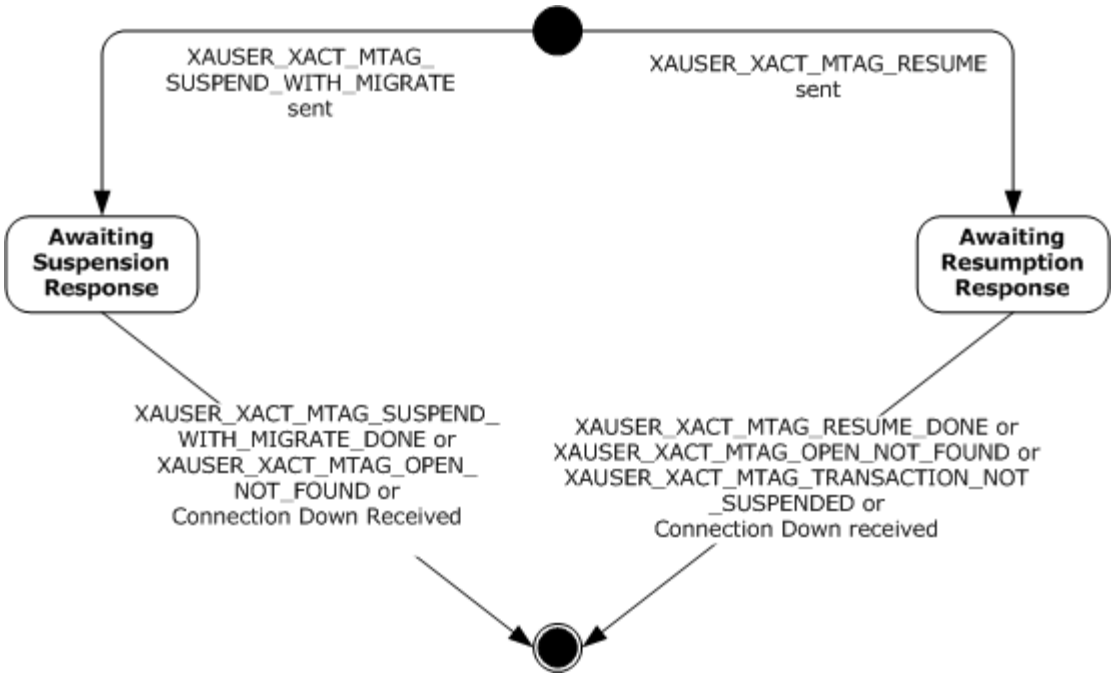


Figure 25: CONNTYPE_ XAUSER_XACT_MIGRATE2 initiator states

3.3.2 Timers

No timers apply here.

3.3.3 Initialization

When the XA Superior Transaction Manager is initialized there are no actions to be taken.

3.3.4 Higher-Layer Triggered Events

3.3.4.1 XA Lookup

The XA Lookup event MUST be signaled with the following arguments.

- XID - The XA Transaction Branch Identifier to look for.
- Resource Manager Identifier - The resource manager ID to be used to look for the corresponding XA Superior Proxy object.

The XA Lookup event MUST return the following value.

- Transaction Identifier - Identifier field of the associated OleTx transaction object. Null indicates a failure.

If the XA Lookup event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- Attempt to find the XA Superior Proxy object whose key is set to the provided Resource Manager Identifier in the XA Superior Proxy Table.

- If the XA Superior Proxy object is found:
  - Attempt to find XA Transaction object whose key is set to the provided XID in the XA Transaction Table referenced by the located XA Superior Proxy object.
    - If the XA Transaction object is found:
      - Return the Transaction Identifier field of the located XA Transaction object.
    - Otherwise:
      - Return NULL.
- Otherwise:
  - Return NULL.

3.3.4.2 Xa_close

The Xa_close event MUST be signaled with the following arguments.

- xa_info ASCII string
  - TM - Transaction Manager description
  - RM Recovery GUID - Resource Manager Recovery GUID
  - Transaction Timeout (optional)
  - Branch Isolation (optional)
- RMID - Resource Manager ID to use for identifying the XA Superior Proxy object in the XA Superior Proxy Table.
- Flags - Flags defined in [XOPEN-DTP] Chapter 4.4.

The xa_close event MUST return the following value.

- XA Result - Return code defined in [XOPEN-DTP] Chapter 4.5.

If the xa_close event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- If the TMASYNC is set in the provided Flags:
  - Return XAER_ASYNC.
- Otherwise, if provided Flags are not equal to TMNOFLAGS:
  - Return E_INVALIDARG.
- Otherwise:
  - Attempt to find an XA Superior Proxy object in the XA Superior Proxy Table corresponding to the provided Resource Manager ID.
  - If the XA Superior Proxy object is found:
    - Decrement the Open Count field of the located XA Superior Proxy object.

- If the Open Count field of the located XA Superior Proxy object is set to 0:
  - Disconnect the transaction manager Connection referenced by the XA Superior Proxy object.
  - Reset the transaction manager Connection referenced by the XA Superior Proxy Object.
  - Remove the XA Superior Proxy object from the XA Superior Proxy Table.
- Otherwise:
  - Return XAER_PROTO.

3.3.4.3 Xa_commit

The Xa_commit event MUST be signaled with the following arguments.

- XID - XA Transaction Branch Identifier
- RMID - Resource Manager ID
- Flags - Flags defined in [XOPEN-DTP]Chapter 4.4.

The Xa_commit event MUST return the following value.

- XA Result- Return code defined in [XOPEN-DTP]Chapter 4.5.

If the Xa_commit event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- If Flags provided has TMASYNC set:
  - Return XAER_ASYNC.
- Otherwise:
  - Attempt to find an XA Superior Proxy object in the XA Superior Proxy Table corresponding to the provided Resource Manager ID.
  - If the XA Superior Proxy object is not found:
    - Return XAER_RMFAIL.
  - Otherwise:
    - Attempt to find an XA Transaction object corresponding to the provided XID in the XA Transaction Table referenced by the located XA Superior Proxy object.
    - If the XA Transaction object is found:
      - Use this as the XA Transaction for the remaining steps.
    - Otherwise:
      - Attempt to create a new XA Transaction Object with the following values.
        - XID field set to the provided XID.
        - Thread Identifier field set to the Current Thread Identifier.

- Migrate field set to FALSE.
- State field set to Suspended.
- The Require Thread Affinity field SHOULD be set to whether Flags does not have TM_NOTHREADAFFINITY set.<21>
- Parent XA Superior Proxy object reference set to located XA Superior Proxy object.

- If failed:
  - Return XAER_RMERR.
- Otherwise:
  - Add this XA Transaction object to the XA Transaction Table referenced by the located XA Superior Proxy object.
  - Use this as the XA Transaction for the remaining steps

- If Branch Isolation field of located XA Superior Proxy object is set to Tight:
  - Attempt to create an XA Transaction CMP Connection with the following settings.
    - Connection Type field set to CONNTYPE_XAUSER_XACT_BRANCH_OPEN.
    - XA Transaction Object set to the located or created XA Transaction object.
    - State field set to Awaiting Open Response.
  - If Successful:
    - Set State field of located or created XA Transaction object to Opening.
    - If provided Flags have TMONEPHASE set:
      - Set Action to Prepare Single Phase.
    - Otherwise:
      - Set Action to Commit.
    - Send an XAUSER_XACT_MTAG_OPEN message with the following values.
      - RMGUID set to Resource Manager Recovery GUID field of located XA Superior Proxy object.
      - UoW Length set to size of an XID.
      - XID set to the provided XID.
  - Otherwise:
    - Return XA_RMFAIL.
- Otherwise:
  - Attempt to create an XA Transaction CMP Connection with the following settings.

    Connection Type field set to CONNTYPE_XAUSER_XACT _OPEN.

XA Transaction Object set to the located or created XA Transaction object.

State field set to Awaiting Open Response.

- If Successful:
  - Set State field of located or created XA Transaction object to Opening.
  - If provided Flags have TMONEPHASE set:
    - Set Action to Prepare Single Phase.
  - Otherwise:
    - Set Action to Commit.
  - Send an XAUSER_XACT_MTAG_OPEN message with the following values.
    - RMGUID set to Resource Manager Recovery GUID field of located XA Superior Proxy object.
    - UoW Length set to size of an XID.
    - XID set to the provided XID.
- Otherwise:
  - Return XA_RMFAIL.

3.3.4.4 Xa_complete

The Xa_complete event MUST be signaled with the following arguments.

- XA Handle - Ignored on receipt.
- XA Return Handle - Ignored on receipt.
- RMID - Resource Manager ID
- Flags - Flags defined in [XOPEN-DTP] Chapter 4.4.

The Xa_complete event MUST return the following value.

- XA Result - Return code defined in [XOPEN-DTP] Chapter 4.5,

If the Xa_complete event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- Return XAER_PROTO.

3.3.4.5 Xa_end

The Xa_end event MUST be signaled with the following arguments.

- XID - XA Transaction Branch Identifier
- RMID - Resource Manager ID
- Flags - Flags defined in Chapter 4.4 of [XOPEN-DTP]

The Xa_end event MUST return the following value.

- XA Result - Return code defined in [XOPEN-DTP] Chapter 4.5.

If the Xa_end event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- If provided Flags has TMASYNC set:
  - Return XAER_ASYNC.
- If provided Flags has TMMIGRATE set:
  - SHOULD check if TMSUSPEND flag is not set:<22>
    - Return XAER_PROTO.
- Attempt to find an XA Superior Proxy object in the XA Superior Proxy Table corresponding to the provided RMID.
- If not found:
  - Return XAER_RMFAIL;
- Otherwise:
  - Attempt to find an XA Transaction object corresponding to the provided XID in the XA Transaction Table referenced by XA Superior Proxy object:
  - If found:
    - If Flags has TMSUSPEND set:
      - If State field of located XA Transaction object is set to Active:
        - Set State field of located XA Transaction to Suspended.
        - If Flags has TMMIGRATE set:
          - Attempt to create XA Transaction CMP Connection with the following settings.
            - Connection type SHOULD be set to CONNTYPE_XAUSER_XACT_MIGRATE2<23>
            - XA Transaction Object set to the located XA Transaction object.
            - State set to Awaiting Resumption Response.
            - XA Transaction object State set to Suspended.
            - Action set to Migrate.
          - If Successful:
            - Send an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE message with the following values.
              - RMGUID set to Resource Manager Identifier field of the located XA Superior Proxy object.

- UoW Length set to size of an XID.
- XID set to provided XID.
- Current Process Identifier.
- Current Thread Identifier.

- Otherwise:
  - Return XA_RMFAIL.

- Otherwise:
  - Return XAER_RMERR.

- Otherwise:
  - Verify that the Thread Identifier field of the XA Transaction object is set to the Current Thread Identifier:
  - If Fail:
    - Return XAER_PROTO.
  - Otherwise:
    - Disconnect Connection referenced by XA Transaction object.
    - Remove the XA Transaction object from the XA Transaction Table of the located XA Superior Proxy.
    - Return XA_OK.

- Otherwise:
  - Return XAER_NOTA.

3.3.4.6 Xa_forget

The Xa_forget event MUST be signaled with the following arguments.

- XID - XA Transaction Branch Identifier
- RMID - Resource Manager ID
- Flags - Flags defined in Chapter 4.4 of [XOPEN-DTP]

The Xa_forget event MUST return the following value.

- XA Result - Return code defined in [XOPEN-DTP] Chapter 4.5.

If the Xa_forget event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- The XA Superior Transaction Manager will return XAER_NOTA.

3.3.4.7 Xa_open

The Xa_open event MUST be signaled with the following arguments.

- xa_info
  - TM - Transaction Manager description
  - RM Recovery GUID - Resource Manager Recovery GUID
  - Transaction Timeout (optional)<24>
  - Branch Isolation (optional)<25>
- RMID - Resource Manager ID
- Flags - Flags defined in [XOPEN-DTP] Chapter 4.4.

The Xa_open event MUST return the following value.

- XA Result - Return code defined in Chapter 4.5 of [XOPEN-DTP]

If the Xa_open event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- If the provided Flags field has TMASYNC set:
  - Return XAER_ASYNC.
- If the provided Flags field is not set to TMNOFLAGS:
  - Return Failure.
- If the XA Transactions Enabled flag is not set:
  - Return XAER_RMERR.
- Attempt to find an XA Superior Proxy object in the XA Superior Table keyed by the provided RMID.
- If an object is found:
  - If the Branch Isolation field of the XA Superior Proxy object is not set to the provided Branch Isolation:
    - Return XAER_INVAL.
  - Otherwise:
    - Increment the Open Count field of the XA Superior Proxy object.
    - If Transaction Timeout is provided:
      - Set the Transaction Timeout field of XA Superior Proxy object to provided Transaction Timeout.
- Otherwise:
  - Attempt to create a new XA Superior Proxy Object with the following values.

- The TM field is set to the TM field of the provided xa_info.
- The Resource Manager Recovery GUID field is set to the provided RM Recovery GUID field.
- Transaction Timeout is set if provided.
- The Branch Isolation field is set to the Branch Isolation field if provided; otherwise set the Branch Isolation field to Loose.
- The Recovery Complete field is set to FALSE.

- If failed:
  - Return XAER_RMERR.
- Otherwise:
  - Attempt to create a new XA Superior Proxy CMP Connection object with the following values.
    - The Connection type field is set to CONNTYPE_XAUSER_CONTROL.
    - The XA Superior Proxy Object is set to created XA Superior Proxy object.
    - The State is set to Awaiting Creation Response.
  - If failed:
    - Return XAER_RMERR.
  - Otherwise:
    - Set the transaction manager Connection field of XA Superior Proxy object to the newly created XA Superior Proxy CMP Connection object.
    - Send an XAUSER_CONTROL_MTAG_CREATE message with the following argument.
      - The **guidXaRm** field of the message is set to the provided RM Recovery GUID.

3.3.4.8 Xa_prepare

The Xa_prepare event MUST be signaled with the following arguments.

- XID - XA Transaction Branch Identifier
- RMID - Resource Manager ID
- Flags - Flags defined in Chapter 4.4 of [XOPEN-DTP]

The Xa_prepare event MUST return the following value.

- XA Result - Return code defined in [XOPEN-DTP] Chapter 4.5.

If the Xa_prepare event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- If the provided Flags field has TMASYNC set:
  - Return XAER_ASYNC.

- Otherwise:
  - Attempt to find an XA Superior Proxy object in the XA Superior Proxy Table keyed by the provided RMID.
  - If an object is not found:
    - Return XAER_RMFAIL.
  - Otherwise:
    - Attempt to find an XA Transaction object in the XA Transaction Table referenced by the XA Superior Proxy object keyed by the provided XID.
    - If an object is found:
      - Use this as the XA Transaction.
    - Otherwise:
      - Attempt to create a new XA Transaction Object with the following values.
        - The XID field is set to the provided XID.
        - The Thread Identifier field is set to the Current Thread ID.
        - The Migrate field is set to FALSE.
        - The State field is set to Suspended.
        - The Require Thread Affinity field SHOULD be set to whether Flags does not have TM_NOTHREADAFFINITY set.<26>
        - The Parent XA Superior Proxy object is set to the found XA Superior Proxy object.
      - If failed:
        - Return XAER_RMERR.
      - Otherwise:
        - Use this as the XA Transaction object.
    - If the found XA Superior Proxy object has the Branch Isolation field set to Tight:
      - Attempt to create a new XA Transaction CMP Connection with the following values.
        - The Connection type field is set to CONNTYPE_XAUSER_XACT_BRANCH_OPEN.
        - The XA Transaction Object is set to the created XA Transaction object.
        - The State field is set to Awaiting Open Response.
        - The State field of the XA Transaction object is set to Opening.
        - The Action field is set to Prepare.
      - Send an XAUSER_XACT_MTAG_OPEN message with the following values.

- The **guidXaRm** is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
- The **lenXAIdentifier** field of XA_UOW is set to size of the provided XID field.
- The **XAIdentifier** field is set to the provided XID field.

- Otherwise:
  - Attempt to create a new XA Transaction CMP Connection object with the following values.
    - The Connection type is set to CONNTYPE_XAUSER_XACT_OPEN.
    - The XA Transaction Object is set to the created XA Transaction object.
    - The State field is set to Awaiting Open Response.
    - The Action field is set to Prepare.
  - Set the State field of the XA Transaction object to Opening.
  - Send an XAUSER_XACT_MTAG_OPEN message with the following values.
    - The **guidXaRm** is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
    - The **lenXAIdentifier** field of XA_UOW is set to size of the provided XID field.
    - The **XAIdentifier** field is set to the provided XID field.

3.3.4.9 Xa_recover

The Xa_recover event MUST be signaled with the following arguments.

- XID Array - Array into which recovered XIDs should be placed.
- Count- Maximum number of XIDs to be returned.
- RMID - Resource Manager ID
- Flags - Flags defined in [XOPEN-DTP] Chapter 4.4.

The Xa_recover event MUST return the following value.

- XA Result - Return code defined in [XOPEN-DTP] Chapter 4.5.

If the Xa_recover event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- If the provided Count field is less than 1:
  - Return XAER_INVAL.
- If the XA Transactions Enabled flag is not set:
  - Return XAER_RMFAIL.
- If the State of receiving Connection is not set to Active:

- Return XAER_RMFAIL.

- Attempt to find an XA Superior Proxy object in the XA Superior Proxy Table keyed by the provided RMID.
- If an object found:
  - Create a RequestFlags variable with no flags set initially.
  - SHOULD test if Recovery Complete field of the found XA Superior Proxy object is set to TRUE and the provided Flags field does not have TMSTARTRSCAN set:<27>
    - Return 0.
  - If TMSTARTRSCAN is set in the provided Flags field:
    - SHOULD set the Recovery Complete field of the found XA Superior Proxy object to FALSE.<28>
    - Set the Recovery Return Number field of the found XA Superior Proxy object to 0.
    - Set XARECOVER_START_SCAN flag in RequestFlags variable.
  - Otherwise:
    - If TMENDRSCAN is set in the provided Flags field:
      - SHOULD set the Recovery Complete field of the found XA Superior Proxy object to TRUE.<29>
      - MAY set XARECOVER_END_SCAN flag in the RequestFlags variable.<30>
    - Otherwise:
      - MAY Set XARECOVER_CONTINUE_SCAN flag in RequestFlags variable.<31>
      - If the provided Flags field is not set to TMNOFLAGS:
        - SHOULD return XAER_RMFAIL.<32>
    - Set XARECOVER_CONTINUE_SCAN flag in RequestFlags variable.
  - SHOULD set the Recovery Request Number field of the found XA Superior Proxy object to a minimum value between an implementation specific value and provided Count.<33>
  - Set the Recovery Max Return field of the found XA Superior Proxy object to the provided Count field.
  - Set the TM Connection's State to Awaiting Recovery Response.
  - Set the Recovery Array field of the found XA Superior Proxy object to reference the provided XID Array.
  - Send an XAUSER_CONTROL_MTAG_RECOVER message with the following values.
    - The **RequestFlags** field is set to Flags to be sent across the wire.
    - The **totalUOWsRequested** field SHOULD be set to a minimum value between an implementation specific value and the provided Count field.<34>

- Otherwise:
  - Return XAER_RMFAIL.

3.3.4.10 Xa_rollback

The Xa_rollback event MUST be signaled with the following arguments.

- XID - XA Transaction Branch Identifier
- RMID - Resource Manager ID
- Flags - Flags defined in Chapter 4.4 of [XOPEN-DTP]

The Xa_rollback event MUST return the following value.

- XA Result - Return code defined in [XOPEN-DTP] Chapter 4.5.

If the Xa_rollback event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- If the provided Flags field has TMASYNC set:
  - Attempt to find an XA Superior Proxy object in the XA Superior Proxy Table keyed by the provided RMID.
  - If an object is not found:
    - Return XAER_RMFAIL.
  - Otherwise:
    - Attempt to find an XA Transaction object in the XA Transaction Table referenced by the XA Superior Proxy object keyed by the provided XID.
    - If an object is found:
      - Use this as the XA Transaction.
    - Otherwise:
      - Attempt to create a new XA Transaction Object with the following values.
        - The XID field is set to the provided XID.
        - The Thread Identifier field is set to the Current Thread ID.
        - The Migrate field is set to FALSE.
        - The State field is set to Suspended.
        - The Require Thread Affinity field SHOULD be set to whether Flags does not have TM_NOTHREADAFFINITY set.<35>
        - The Parent XA Superior Proxy object is set to XA Superior Proxy object.
      - If failed:
        - Return XAER_RMERR.

- Otherwise:
  - Use this as the XA Transaction object.

- If the found XA Superior Proxy object has Branch Isolation field set to Tight:
  - Create a new XA Transaction CMP Connection object with the following values.
    - The Connection type field is set to CONNTYPE_XAUSER_XACT_BRANCH_OPEN.
    - The XA Transaction Object is set to created XA Transaction object.
    - The State field set to Awaiting Open Response.
    - The State field of the XA Transaction object is set to Opening.
    - The Action field is set to Rollback.
  - Send an XAUSER_XACT_MTAG_OPEN message with the following values.
    - The **guidXaRm** is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
    - The **lenXAIdentifier** field of XA_UOW is set to size of the provided XID field.
    - The **XAIdentifier** field is set to the provided XID field.
- Otherwise:
  - Attempt to create a new XA Transaction CMP Connection object with the following values.
    - The Connection type field is set to CONNTYPE_XAUSER_XACT_OPEN.
    - The XA Transaction object is set to the created XA Transaction object.
    - The State field is set to Awaiting Open Response.
    - The Action field is set to Rollback.
  - The State field of the XA Transaction object is set to Opening.
  - Send an XAUSER_XACT_MTAG_OPEN message with the following values.
    - The **guidXaRm** is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
    - The **lenXAIdentifier** field of XA_UOW is set to size of the provided XID field.
    - The **XAIdentifier** field is set to the provided XID field.

3.3.4.11 Xa_start

The Xa_start event MUST be signaled with the following arguments.

- XID - XA Transaction Branch ID
- RMID - Resource Manager ID

- Flags - Flags defined in [XOPEN-DTP] Chapter 4.4.

The Xa_start event MUST return the following value.

- XA Result - Return code defined in [XOPEN-DTP] Chapter 4.5.

If the Xa_start event is signaled, the XA Superior Transaction Manager MUST perform the following actions.

- If the provided Flags field has TMASYNC set:
  - Return XAER_ASYNC.
- Attempt to find an XA Superior Proxy object in the XA Superior Proxy Table keyed by the provided RMID.
- If an object is not found:
  - Return XAER_RMFAIL.
- Otherwise:
  - Attempt to find an XA Transaction object in the XA Transaction Table referenced by the found XA Superior Proxy object keyed by the provided XID.
  - If an object is found:
    - If the provided Flags field has TMRESUME set:
      - If the State field of the found XA Transaction object is set to Suspended:
        - If the Migrate field of the found XA Transaction object is set to TRUE:
          - Attempt to create a new XA Transaction CMP Connection object with the following values.
            - The Connection type field SHOULD be set to CONNTYPE_XAUSER_XACT_MIGRATE2.<36>
            - The XA Transaction Object is set to the located XA Transaction object.
            - The State field is set to Awaiting Resumption Response.
            - The State field of the XA Transaction object is set to Suspended.
            - The Action field is set to Resume.
          - Send an XAUSER_XACT_MTAG_RESUME message with the following values.
            - The **guidXaRm** field is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
            - The **lenXAIdentifier** field is set to the size of an XID.
            - The **XAIdentifier** field is set to the provided XID.
            - The **dwProcessId** field is set to the Current Process ID.
            - The **dwThreadId** field is set to the Current Thread ID.

- Otherwise:
  - Set the State of the found XA Transaction object to Active.
- Otherwise:
  - Return XAER_RMERR.
- Otherwise:
  - If the provided Flags field does not have TMJOIN set:
    - Return XAER_DUPID.
  - Otherwise:
    - If the Require Thread Affinity field of the found XA Transaction object is set to TRUE, check if the Current Thread ID is set to the Thread Identifier of the found XA Transaction object.
    - If failed:
      - Return XAER_RMERR.
    - Otherwise:
      - If the State field of the found XA Transaction object is set to Suspended:
        - SHOULD set the State field of the XA Transaction object to Active.<37>
      - Otherwise:
        - Return XAER_RMERR.
- Otherwise:
  - If the provided Flags have TMRESUME set:
    - MAY return XAER_RMERR .<38>
    - Attempt to create a new XA Transaction object with the following values.
      - The XID field is set to the provided XID.
      - The Thread Identifier field is set to the Current Thread ID.
      - The Migrate field is set to FALSE.
      - The State field is set to Suspended.
      - The Require Thread Affinity field SHOULD be set to whether Flags does not have TM_NOTHREADAFFINITY set.<39>
      - The Parent XA Superior Proxy object is set to the found XA Superior Proxy object.
    - If failed:
      - Return XAER_RMERR.
    - Otherwise:

- Create XA Superior Enlistment CMP Connection object with the following values.
  - The Connection type field SHOULD be set to CONNTYPE_XAUSER_XACT_MIGRATE2 .<40>
  - The XA Superior Enlistment object is set to created XA Superior Enlistment object.
  - The State field is set to Awaiting Resumption Response.
  - The State field of the created XA Transaction object is set to Suspended.
  - The Action field is set to Resume.
- Set the State field of the XA Superior Enlistment object Suspended.
- Send an XAUSER_XACT_MTAG_RESUME message with the following values.
  - The **guidXaRm** field is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
  - The **lenXAIdentifier** field is set to the size of an XID.
  - The **XAIdentifier** field is set to the provided XID.
  - The **dwProcessId** field is set to the Current Process ID.
  - The **dwThreadId** field is set to the Current Thread ID.

- Otherwise:
  - Attempt to create a new XA Transaction object with the following values.
    - The **XID** field is set to the provided XID.
    - The **Thread Identifier** field is set to the Current Thread ID.
    - The **Migrate** field is set to FALSE.
    - The **State** field is set to Idle.
    - The Require Thread Affinity field SHOULD be set to whether Flags does not have TM_NOTHREADAFFINITY set.<41>
    - The Parent XA Superior Proxy object is set to the found XA Superior Proxy object.
  - If failed:
    - Return XAER_RMERR.
  - Otherwise:
    - If the found XA Superior Proxy object has the Branch Isolation field set to Tight:
      - If the provided Flags has TMJOIN set:
        - Create a new XA Transaction CMP Connection object with the following values.
          - The Connection type is set to CONNTYPE_XAUSER_XACT_BRANCH_OPEN.

- The XA Transaction object is set to the located created XA Transaction object.
- The State field is set to Awaiting Open Response.
- The State field of the XA Transaction object is set to Opening.
- The Action field is set to None.

- Set the State field of the XA Superior Enlistment object to Opening.
- Send an XAUSER_XACT_MTAG_OPEN message with the following values.
  - The **guidXaRm** is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
  - The **lenXAIdentifier** field of XA_UOW is set to size of the provided XID field.
  - The **XAIdentifier** field is set to the provided XID field.

- Otherwise:
  - Create a new XA Transaction CMP Connection object with the following values.
    - The Connection Type is set to CONNTYPE_XAUSER_XACT_BRANCH_START.
    - The XA Transaction object is set to the located XA Transaction object.
    - The State field of the XA Transaction object is set to Starting.
    - The State field is set to Awaiting Start Response.
    - The Action field is set to None.
  - Send an XAUSER_XACT_MTAG_START message with the following values.
    - The **guidXaRm** field is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
    - The **lenXAIdentifier** field of XA_UOW is set to size of the provided XID field.
    - The **XAIdentifier** field is set to the provided XID field.
    - The **isoLevel** field is set to ISOLATIONLEVEL_ISOLATED.
    - The **Timeout** field is set to the Transaction Timeout field of the found XA Superior Proxy object.
    - If the TM field of the found XA Superior Proxy is not set to "":
      - Set the **szDesc** field of the message "Transaction", appended if possible.
    - Otherwise:
      - Set the **szDesc** field of the message to "XA Transaction".
    - The **isoFlags** field is set to 0.

- Otherwise:
  - If the provided Flags has TMJOIN set:
    - Create a new XA Transaction CMP Connection object with the following values.
      - The Connection type is set to CONNTYPE_XAUSER_XACT_OPEN.
      - The XA Transaction object is set to the located XA Transaction object.
      - The State is set to Awaiting Open Response.
      - The Action is set to None.
    - Set the State of the created XA Transaction object to Opening.
    - Set the State of the created XA Superior Enlistment object to Opening.
    - Send an XAUSER_XACT_MTAG_OPEN message with the following values.
      - The **guidXaRm** is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
      - The **lenXAIdentifier** field of XA_UOW is set to size of the provided XID field .
      - The **XAIdentifier** field is set to the provided XID field.
  - Otherwise:
    - Create a new XA Transaction CMP Connection object with the following values.
      - The Connection type is set to CONNTYPE_XAUSER_XACT_START.
      - The XA Transaction object is set to the located XA Transaction object.
      - The **State** field is set to Awaiting Start Response.
      - The **Action** field is set to None.
    - The State of the created XA Transaction object is set to Starting.
    - Set the State of the created XA Superior Enlistment object to Starting.
    - Send an XAUSER_XACT_MTAG_START message with the following values.
      - The **guidXaRm** field is set to the Resource Manager Recovery GUID field of the found XA Superior Proxy object.
      - The **lenXAIdentifier** field of XA_UOW is set to size of the provided XID field.
      - The **XAIdentifier** field is set to the provided XID field.
      - The **isoLevel** field is set to ISOLATIONLEVEL_ISOLATED.
      - The **Timeout** field is set to the Transaction Timeout field of the found XA Superior Proxy object.

- If the TM field of the found XA Superior Proxy is not set to "":
  - Set the **szDesc** field of the message to "Transaction", appended if possible.
- Otherwise:
  - Set the **szDesc** field of the message to "XA Transaction".
- The **isoFlags** field is set to 0.

3.3.5 Message Processing Events and Sequencing Rules

3.3.5.1 CONNTYPE_XAUSER_CONTROL Initiator

3.3.5.1.1 Receiving an XAUSER_CONTROL_MTAG_CREATE_NO_MEM Message

When the XA Superior Transaction Manager receives the following:

- XAUSER_CONTROL_MTAG_CREATE_NO_MEM
- Connection Down
- Connection Disconneted

it MUST perform the following actions.

- If the State field of the receiving XA Superior Proxy CMP Connection is set to Creating:
  - Return XAER_RMERR from xa_open call.
  - Disconnect the receiving connection.
  - Reset the transaction manager Connection field of XA Superior Proxy object referenced by receiving connection.
- Otherwise:
  - Disconnect receiving connection.
  - Reset the transaction manager Connection field of XA Superior Proxy object referenced by receiving connection.

3.3.5.1.2 Receiving an XAUSER_CONTROL_MTAG_CREATED Message

When the XA Superior Transaction Manager receives an XAUSER_CONTROL_MTAG_CREATED message, it MUST perform the following actions.

- If the State field of receiving XA Superior Proxy CMP Connection is set to Creating:
  - Set the State field of receiving XA Superior Proxy CMP Connection to Active.
  - Add the XA Superior Proxy object referenced by receiving connection to XA Superior Proxy Table.
  - Return XA_OK from xa_open call.
- Otherwise:

- Disconnect the receiving connection.
- Reset the transaction manager Connection field of XA Superior Proxy object referenced by receiving connection.

3.3.5.1.3 Receiving an XAUSER_CONTROL_MTAG_RECOVER_NO_MEM Message

When the XA Superior Transaction Manager receives the following message:

- XAUSER_CONTROL_MTAG_RECOVER_NO_MEM

it MUST perform the following actions.

- If the State field of the receiving XA Superior Proxy CMP Connection is set to Recovering:
  - Return XAER_RMFAIL from xa_recover call.
  - Disconnect the receiving connection.
  - Reset the transaction manager Connection field of XA Superior Proxy object referenced by the receiving connection.
  - Remove the XA Superior Proxy object referenced by receiving connection from XA Superior Proxy Table.
- Otherwise:
  - Disconnect the receiving connection.
  - Reset the transaction manager Connection field of XA Superior Proxy object referenced by receiving connection.
  - Remove the XA Superior Proxy object referenced by receiving connection from XA Superior Proxy Table.

3.3.5.1.4 Receiving an XAUSER_CONTROL_MTAG_RECOVER_REPLY Message

When the XA Superior Transaction Manager receives an XAUSER_CONTROL_MTAG_RECOVER_REPLY message, it MUST perform the following actions.

- If the State of receiving Connection is not set to Recovering:
  - Disconnect the receiving connection.
  - Reset the transaction manager Connection field of XA Superior Proxy object referenced by the receiving connection.
  - Remove the XA Superior Proxy object referenced by receiving connection from the XA Superior Proxy Table.
- If the **ulTotalUoWs** field of the message is greater than Recovery Request Number, SHOULD perform the following actions.<42>
  - Return XAER_RMFAIL from xa_recover call.
  - Disconnect the receiving connection.

- Reset the transaction manager Connection field of XA Superior Proxy object referenced by the receiving connection.
- Remove the XA Superior Proxy object referenced by the receiving connection from XA Superior Proxy Table.

- If the **ReplyFlags** field of the message has XARECOVER_END_OF_RECS set:
  - Set the Recovery Complete field of XA Superior Proxy object referenced by receiving connection.
  - Copy the **ulTotalUoWs** field of the message number of XIDs from **UoW_Recs** field of the message to Recovery Array of the XA Superior Proxy object referenced by the receiving connection.
  - Increment the Recovery Return Number field of the XA Superior Proxy object referenced by the receiving connection by **ulTotalUoWs** field of the message.
  - If the Recovery Return Number field of the XA Superior Proxy object referenced by the receiving connection is set to the value of Recovery Max Return field of the XA Superior Proxy object referenced by the receiving connection or **ulTotalUoWs** field of the message is less than Recovery Request Number field of the XA Superior Proxy object referenced by the receiving connection:
    - Set the State of the receiving connection to Active.
    - Return Recovery Return Number Return Number field of the XA Superior Proxy object referenced by the receiving connection from xa_recover.
  - Otherwise:
    - Set Recovery Request Number field of the XA Superior Proxy object referenced by the receiving connection to minimum value between an implementation specific value and the difference between the Recovery Max Return field of the XA Superior Proxy Object referenced by the receiving connection and the Recovery Return Number field of the XA Superior Proxy Object referenced by the receiving connection.<43>
    - Send an XAUSER_CONTROL_MTAG_RECOVER message with the following values.
      - The **RequestFlags** field is set to XARECOVER_CONTINUE_SCAN.
      - The **totalUOWsRequested** field is set to Recovery Request Number on XA Superior Proxy object referenced by the receiving connection.

3.3.5.1.5 Connection Down, Connection Disconnected

When the XA Superior Transaction Manager receives the following:

- Connection Down
- Connection Disconnected

it MUST perform the following actions.

- If the State field of the receiving XA Superior Proxy CMP Connection is set to Creating:
  - Return XAER_RMERR from xa_open call.

- Disconnect the receiving connection.
- Reset the transaction manager Connection field of XA Superior Proxy object referenced by receiving connection.

- Otherwise, if the State field of the receiving XA Superior Proxy CMP Connection is set to Recovering:
  - Return XAER_RMFAIL from xa_recover call.
  - Disconnect the receiving connection.
  - Reset the transaction manager Connection field of the XA Superior Proxy object referenced by the receiving connection.
  - Remove the XA Superior Proxy object referenced by receiving connection from XA Superior Proxy Table.
- Otherwise:
  - Disconnect receiving connection.
  - Reset the transaction manager Connection field of XA Superior Proxy object referenced by receiving connection.

3.3.5.2 CONNTYPE_XAUSER_XACT_START Initiator

3.3.5.2.1 Receiving an XAUSER_XACT_MTAG_STARTED Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_STARTED message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection object is set to Awaiting Start Response:
  - Set the Transaction Identifier of XA Transaction object referenced by receiving connection to the **guidTx** field of the message.
  - Set the State of the receiving connection to Active.
  - Set the State of XA Transaction object referenced by receiving connection to Active.
  - Add the XA Transaction object referenced by receiving connection to XA Transaction Table referenced by the Parent XA Superior Proxy referenced by the XA Transaction object.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.2.2 Receiving an XAUSER_XACT_MTAG_START_NO_MEM Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_START_NO_MEM message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection is set to Awaiting Start Response:
  - Return XAER_RMERR from xa_start call.

- Disconnect the receiving connection.

- Otherwise:

  - Disconnect the receiving connection.

3.3.5.2.3 Receiving an XAUSER_XACT_MTAG_START_LOG_FULL Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_START_LOG_FULL message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection is set to Awaiting Start Response:

  - Return XAER_RBTRANSIENT from xa_start call.

  - Disconnect the receiving connection.

- Otherwise:

  - Disconnect the receiving connection.

3.3.5.2.4 Receiving an XAUSER_XACT_MTAG_START_DUPLICATE Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_START_DUPLICATE message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection is set to Awaiting Start Response:

  - Return XAER_DUPID from xa_start call.

  - Disconnect the receiving connection.

- Otherwise:

  - Disconnect the receiving connection.

3.3.5.2.5 Connection Down, Connection Disconnected

When CONNTYPE_XAUSER_XACT_START connection is disconnected, XA Superior Transaction Manager MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection is set to Awaiting Start Response:

  - Return XAER_RMFAIL from xa_start call.

  - Disconnect the receiving connection.

- Otherwise:

  - Disconnect the receiving connection.

3.3.5.3 CONNTYPE_XAUSER_XACT_OPEN Initiator

3.3.5.3.1 Receiving an XAUSER_XACT_MTAG_OPENED Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_OPENED message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection is set to Awaiting Open Response:
  - Set the Transaction Identifier field of the XA Transaction object referenced by receiving connection to the **guidTx** field of the message.
  - Set the receiving connection state to Active.
  - Set the State field of the XA Transaction object referenced by receiving connection to Active.
  - Add the XA Transaction Object referenced by receiving connection to the XA Transaction Table referenced by the Parent XA Superior Proxy referenced by the XA Transaction object.
  - If the Action field of the receiving Connection is set to Prepare:
    - Set the State field of the receiving connection to Preparing.
    - Set the State field of XA Transaction object referenced by receiving connection to Preparing.
    - Send an XAUSER_XACT_MTAG_PREPARE message with the following values.
      - The **fSinglePhase** field set to FALSE.
  - Otherwise if the Action field of the receiving Connection is set to Prepare Single Phase:
    - Set the State field of the receiving connection to Preparing.
    - Set the State field of the XA Transaction object referenced by receiving connection to Preparing.
    - Send an XAUSER_XACT_MTAG_PREPARE message with the following values.
      - The **fSinglePhase** field set to TRUE.
  - Otherwise if the Action field of the receiving Connection is set to Commit:
    - Set the State of the receiving connection to Committing.
    - Set the State field of the XA Transaction object referenced by receiving connection to Committing.
    - Send an XAUSER_XACT_MTAG_ABORT message.
  - Otherwise if the Action field of the receiving Connection is set to Rollback:
    - Set the State field of the receiving connection to Aborting.
    - Set the State field of the XA Transaction object referenced by receiving connection to Aborting.
    - Send an XAUSER_XACT_MTAG_ABORT message.

- Otherwise:
  - Disconnect the receiving connection.

3.3.5.3.2 Receiving an XAUSER_XACT_MTAG_OPEN_NOT_FOUND Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message, it MUST perform the following actions.

- If the State of receiving XA Transaction CMP Connection is set to Awaiting Open Response:
  - Return XAER_NOTA from originating XA call.
  - Disconnect the receiving connection.

3.3.5.3.3 Receiving an XAUSER_XACT_MTAG_REQUEST_COMPLETED Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_REQUEST_COMPLETED message, it MUST perform the following actions.

- If the State field of the receiving Connection is set to Preparing:
  - Set the State field of the XA Transaction object to Prepared.
  - Disconnect the receiving Connection.
  - Return XA_OK to xa_prepare call.
- If the State field of the receiving Connection is set to Committing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_OK to xa_commit call.
- If the State field of the receiving Connection is set to Aborting:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_OK to xa_rollback call.

3.3.5.3.4 Receiving an XAUSER_XACT_MTAG_PREPARE_ABORT Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_PREPARE_ABORT message, it MUST perform the following actions.

- If Receiving connection state is Preparing:
  - Set the State field of the XA Transaction object to Complete.

- Disconnect the receiving Connection.
- Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
- Return XA_RBROLLBACK to xa_prepare.

3.3.5.3.5 Receiving an XAUSER_XACT_MTAG_PREPARE_SINGLEPHASE_INDOUBT Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_PREPARE_SINGLEPHASE_INDOUBT message, it MUST perform the following actions.

- If the State field of the receiving connection is set to Preparing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_RBPROTO to xa_prepare.

3.3.5.3.6 Receiving an XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message, it MUST perform the following actions.

- If the State field of the receiving connection is set to Preparing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XAER_PROTO to xa_prepare.
- If the State field of the receiving connection is set to Committing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XAER_PROTO to xa_commit call.
- If the State field of the receiving connection is set to Aborting:
  - Set the State field of the XA Transaction object to Complete.

- Disconnect the receiving Connection.
- Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
- Return XAER_PROTO to xa_rollback call.

3.3.5.3.7 Connection Down, Connection Disconnected

When CONNTYPE_XAUSER_XACT_OPEN connection is disconnected, the XA Superior Transaction Manager MUST perform the following actions,

- If the State field of the receiving connection is set to Preparing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_RBCOMMFAIL to xa_prepare.
- If the State field of the receiving connection is set to Committing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - SHOULD return XAER_RMFAIL to xa_commit call.<44>
- If the State field of the receiving connection is set to Aborting:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - SHOULD return XAER_RMFAIL to xa_rollback call.<45>

3.3.5.4 CONNTYPE_XAUSER_XACT_MIGRATE Initiator

3.3.5.4.1 Receiving an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE message, it MUST perform the following actions.

- If the State field of the XA Transaction object referenced by the receiving connection is set to Suspended:
  - Set the Migrate field of the XA Transaction object to TRUE.

- Return XA_OK to xa_end.

3.3.5.4.2 Receiving an XAUSER_XACT_MTAG_RESUME_DONE Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_RESUME_DONE message, it MUST perform the following actions.

- If the State field of the XA Transaction object referenced by the receiving connection is set to Suspended:
  - Set the State field of the XA Transaction object referenced by the receiving connection to Active.
  - Insert the XA Transaction object in the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_OK to xa_start call.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.4.3 Receiving an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED message, it MUST perform the following actions.

- If the State field of the XA Transaction object referenced by the receiving connection is set to Suspended:
  - Disconnect the receiving connection.
  - Return XAER_PROTO to xa_start call.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.4.4 Receiving an XAUSER_XACT_MTAG_OPEN_NOT_FOUND Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message, it MUST perform the following actions.

- If the State field of the XA Transaction object referenced by the receiving connection is set to Suspended:
  - Disconnect the receiving connection.
  - Return XAER_NOTA to xa_start call.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.4.5 Receiving an XAUSER_XACT_MTAG_START_NO_MEM Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_START_NO_MEM message, it MUST perform the following actions.

- If the State field of the XA Transaction object referenced by the receiving connection is set to Suspended:
  - Disconnect the receiving connection.
  - Return XAER_RMERR to xa_start call.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.5 CONNTYPE_XAUSER_XACT_BRANCH_START Initiator

3.3.5.5.1 Receiving an XAUSER_XACT_MTAG_STARTED Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_STARTED message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection object is set to Awaiting Start Response:
  - Set the Transaction Identifier field of the XA Transaction object referenced by receiving connection to the **guidTx** field of the message.
  - Set the State field of the receiving connection to Active.
  - Set the State field of the XA Transaction object referenced by receiving connection to Active.
  - Add the XA Transaction Object referenced by the receiving connection to the XA Transaction Table, referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.5.2 Receiving an XAUSER_XACT_MTAG_START_NO_MEM Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_START_NO_MEM message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection object is set to Awaiting Start Response:
  - Return XAER_RMERR from xa_start call.
  - Disconnect the receiving connection.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.5.3 Receiving an XAUSER_XACT_MTAG_START_LOG_FULL Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_START_LOG_FULL message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection object is set to Awaiting Start Response:
  - Return XAER_RBTRANSIENT from xa_start call.
  - Disconnect the receiving connection.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.5.4 Receiving an XAUSER_XACT_MTAG_START_DUPLICATE Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_START_DUPLICATE message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection object is set to Awaiting Start Response:
  - Return XAER_DUPID from xa_start call.
  - Disconnect the receiving connection.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.5.5 Connection Down, Connection Disconnected

When CONNTYPE_XAUSER_XACT_BRANCH_START connection is disconnected, XA Superior Transaction Manager MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection object is set to Awaiting Start Response:
  - Return XAER_RMFAIL from xa_start call.
  - Disconnect the receiving connection.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.6 CONNTYPE_XAUSER_XACT_BRANCH_OPEN Initiator

3.3.5.6.1 Receiving an XAUSER_XACT_MTAG_OPENED Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_OPENED message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection object is set to Awaiting Open Response:

- Set Transaction Identifier of XA Transaction object referenced by receiving connection to the **guidTx** field of the message.
- Set the State of the receiving Connection to Active.
- Set the State field of the XA Transaction object referenced by receiving connection to Active.
- Add the XA Transaction object referenced by receiving connection to the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
- If the Action field of the receiving Connection is set to Prepare:
  - Set the State field of the receiving connection to Preparing.
  - Set the State field of the XA Transaction object referenced by receiving connection to Preparing.
  - Send an XAUSER_XACT_MTAG_PREPARE message with the following values.
    - The **fSinglePhase** field is set to FALSE.
- Otherwise, if the Action field of the receiving Connection is set to Prepare Single Phase:
  - Set the State of the receiving connection to Preparing.
  - Set the State field of the XA Transaction object referenced by receiving connection to Preparing.
  - Send an XAUSER_XACT_MTAG_PREPARE message with the following values.
    - The **fSinglePhase** field is set to TRUE.
- Otherwise, if the Action field of the receiving Connection is set to Commit:
  - Set the State of the receiving connection to Committing.
  - Set the State field of the XA Transaction object referenced by receiving connection to Committing.
  - Send an XAUSER_XACT_MTAG_ABORT message.
- Otherwise, if the Action field of the receiving Connection is set to Rollback:
  - Set the State of the receiving connection to Aborting.
  - Set State of XA Transaction object referenced by the receiving Connection to Aborting.
  - Send an XAUSER_XACT_MTAG_ABORT message.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.6.2 Receiving an XAUSER_XACT_MTAG_OPEN_NOT_FOUND Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_OPEN_NOT_FOUND message, it MUST perform the following actions.

- If the State field of the receiving XA Transaction CMP Connection object is set to Awaiting Open Response:
  - Return XAER_NOTA from originating XA call.
  - Disconnect the receiving connection.

3.3.5.6.3 Receiving an XAUSER_XACT_MTAG_REQUEST_COMPLETED Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_REQUEST_COMPLETED message, it MUST perform the following actions.

- If the State field of the receiving Connection is set to Preparing:
  - Set the State field of the XA Transaction object to Prepared.
  - Disconnect the receiving Connection.
  - Return XA_OK to xa_prepare call.
- If the State field of the receiving Connection is set to Committing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_OK to xa_commit call.
- If the State field of the receiving Connection is set to Aborting:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_OK to xa_rollback call.

3.3.5.6.4 Receiving an XAUSER_XACT_MTAG_PREPARE_ABORT Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_PREPARE_ABORT message, it MUST perform the following actions.

- If the State field of the receiving Connection is set to Preparing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_RBROLLBACK to xa_prepare.

3.3.5.6.5 Receiving an XAUSER_XACT_MTAG_PREPARE_SINGLEPHASE_INDOUBT Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_PREPARE_SINGLEPHASE_INDOUBT message, it MUST perform the following actions.

- If the State field of the receiving Connection is set to Preparing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_RBPROTO to xa_prepare.

3.3.5.6.6 Receiving an XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_REQUEST_FAILED_BAD_PROTOCOL message, it MUST perform the following actions.

- If the State field of the receiving Connection is set to Preparing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XAER_PROTO to xa_prepare.
- If the State field of the receiving Connection is set to Committing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XAER_PROTO to xa_commit call.
- If the State field of the receiving Connection is set to Aborting:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XAER_PROTO to xa_rollback call.

3.3.5.6.7 Receiving an XAUSER_XACT_MTAG_READONLY Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_READONLY message, it MUST perform the following actions.

- If the State field of the receiving Connection is set to Preparing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_READONLY to xa_prepare.

3.3.5.6.8 Connection Down, Connection Disconnected

When the CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection is disconnected, the XA Superior Transaction Manager MUST perform the following actions.

- If the State field of the receiving Connection is set to Preparing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Return XA_RBCOMMFAIL to xa_prepare.
- If the State field of the receiving Connection is set to Committing:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - SHOULD return XAER_RMFAIL to xa_commit call.<46>
- If the State field of the receiving Connection is set to Aborting:
  - Set the State field of the XA Transaction object to Complete.
  - Disconnect the receiving Connection.
  - Remove the XA Transaction object from the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - SHOULD return XAER_RMFAIL to xa_rollback call.<47>

3.3.5.7 CONNTYPE_XAUSER_XACT_MIGRATE2 Initiator

3.3.5.7.1 Receiving an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_SUSPEND_WITH_MIGRATE_DONE message, it MUST perform the following actions.

- If the State field of the XA Transaction object referenced by receiving connection is set to Suspended:
  - Set the Migrate field of the XA Transaction object to TRUE.
  - Return XA_OK to xa_end.

3.3.5.7.2 Receiving an XAUSER_XACT_MTAG_RESUME_DONE Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_RESUME_DONE message, it MUST perform the following actions.

- If the State field of the XA Transaction object referenced by receiving connection is set to Suspended:
  - Set the Transaction Identifier field of the XA Transaction referenced by receiving connection to the Transaction Identifier field of the message.
  - Insert the XA Transaction object into the XA Transaction Table referenced by the Parent XA Superior Proxy object referenced by the XA Transaction object.
  - Set the State field of the XA Transaction object to Active.
  - Return XA_OK to xa_start call.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.7.3 Receiving an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_TRANSACTION_NOT_SUSPENDED message, it MUST perform the following actions.

- If the State of XA Transaction referenced by the receiving connection is Suspended:
  - Disconnect receiving connection.
  - Return XAER_PROTO to xa_start call.
- Otherwise:
  - Disconnect receiving connection.

3.3.5.7.4 Connection Down, Connection Disconnected

When the CONNTYPE_XAUSER_XACT_MIGRATE2 connection is disconnected, the XA Superior Transaction Manager MUST perform the following actions.

- If the State field of the XA Transaction object referenced by the receiving connection is set to Suspended:
  - Disconnect the receiving connection.
  - Return XAER_NOTA to xa_start call.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.7.5 Receiving an XAUSER_XACT_MTAG_START_NO_MEM Message

When the XA Superior Transaction Manager receives an XAUSER_XACT_MTAG_START_NO_MEM message, it MUST perform the following actions.

- If the State field of the XA Transaction object referenced by the receiving connection is set to Suspended:
  - Disconnect the receiving connection.
  - Return XAER_RMERR to xa_start call.
- Otherwise:
  - Disconnect the receiving connection.

3.3.5.7.6 Receiving an MTAG_CONNECTION_REQ_DENIED Message

When the XA Superior Transaction Manager receives an MTAG_CONNECTION_REQ_DENIED message, it MUST perform the following actions.

- If the State field of the XA Transaction Object referenced by receiving connection is set to Suspended:
  - Set the Connection Type field of the XA Transaction CMP Connection object to CONNTYPE_XAUSER_XACT_MIGRATE.
  - Send an XAUSER_XACT_MTAG_RESUME message with the following values.
    - The RMGUID set to the Resource Manager Recovery GUID field of the Parent XA Superior Proxy Object referenced by the XA Transaction Object referenced by the receiving connection.
    - The UoW Length set to size of an XID.
    - The XID field is set to XID field of the message.
    - The Current Process Identifier.
    - The Current Thread Identifier.

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

3.4 XA Resource Manager Bridge Facet Details

3.4.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate the implementations adhere to this model as long as their external behavior is consistent with the behavior that is described in this document.

Note that the abstract data model can be implemented in a variety of ways. This protocol does not prescribe or advocate any specific implementation technique.

The XA Resource Manager Bridge Facet MUST maintain all the data elements that are specified in section 3.1.1.

The XA Resource Manager Bridge Facet MUST also maintain the following data elements.

- XA Resource Manager Durable Log: A durable list of XA Resource Manager objects. The contents of this log MUST persist across software restarts or transient failures.
- XA Resource Manager Table: A table of all currently active XA Resource Manager objects.
- XA Resource Manager Identifier Index: A monotonically increasing counter to provide Resource Manager Identifiers that are unique for the lifetime of an XA Resource Manager.
- XaTmMinWarmRecoveryInterval: Specifies minimum time in seconds to wait between Recovery attempts. After each successive failure the wait will be doubled to a maximum of XaTmMaxWarmRecoveryInterval.
- XaTmMaxWarmRecoveryInterval: Specifies maximum time in seconds to wait between Recovery attempts.
- XA Resource Manager object: Represents a currently active XA Resource Manager. This extends the resource manager object defined in [MS-DTCO].
  - When an XA Resource Manager object is stored in the XA Resource Manager Durable Log, the XA Resource Manager Bridge Facet MUST record only the following fields.
    - Resource Manager Global Identifier
    - Data Source Name
    - XA DLL Name
  - This object MUST contain the following elements.
    - **Resource Manager Global Identifier:** Unique identifier associated with XA Resource Manager. Persists through failure and recovery.

- **Data Source Name:** Implementation specific ASCII string to be passed to xa_open and xa_close calls on the XA Switch.
- **Resource Manager Identifier:** Integer identifier to be passed to XA calls. This identifier is unique for the lifetime of the XA Resource Manager instance.
  - **State:** Specifies the current state of XA Resource Manager object. This field MUST contain one of the following values.
  - **Idle:** This is the initial state.
  - **Recovering:** The Recovery processing is being carried out.
  - **Active:** The XA Resource Manager object has been registered.
  - **Ended:** The XA Resource Manager object has been unregistered.
- **XA Switch:** Interface to XA Resource Manager. For a fuller definition, see [XOPEN-DTP] Chapter 5.
- **Recovery Interval:** Specifies the interval for Recovery timer.
- **XA DLL Name:** Specifies DLL Name used to look up XA Switch for XA Resource Manager.
- **Single Pipe:** A flag used to indicate whether XA Resource is Single Pipe or Two Pipe.
- **XA Subordinate Enlistment Table:** A table of currently existing XA Subordinate Enlistment objects in the XA Resource Manager object.
- **Call Count:** Specifies number of calls made to register an XA Resource Manager object with a given Data Source Name.
- **Request Connections Table:** A table containing the XA Resource Manager CMP Connection objects associated with the XA Resource Manager object.
- **Pending Open Connections Table:** A Table containing the XA Resource Manager CMP Connection objects associated with registration requests received during recovery.

- **XA Subordinate Enlistment object:** Represents a XA Subordinate Enlistment to an OleTx transaction. This extends the Enlistment object specified in [MS-DTCO]. This object MUST contain the following elements.
  - **State:** Specifies the current state of XA Subordinate Enlistment object. This field MUST contain one of the following values.
    - **Idle:** This is the initial state.
    - **Enlisted:** XA Subordinate Enlistment object is enlisted in a transaction object.
    - **Prepared:** State field of the Transaction object associated with the XA Subordinate Enlistment object is set to Prepared.
    - **Ended:** The processing for the XA Subordinate Enlistment object has been completed.
  - **Resource Manager Identifier:** Specifies the Identifier for the XA Resource Manager.
  - XID: Specifies the XA Transaction Identifier associated with the Enlistment.

- **Current Request Connection:** Specifies the CMP Connection on which the Enlistment request was received.
- **XA Resource Manager:** Specifies the reference to the XA Resource Manager object corresponding to the XA Subordinate Enlistment object.

- **XA Resource Manager CMP Connection object:** CMP Connection associated with CONNTYPE_XATM_OPEN and CONNTYPE_XATM_OPENONEPIPE acceptors. The definition of an [MS-CMP] connection object is extended to include the following element.
  - Reference to XA Resource Manager object
- **XA Subordinate Enlistment CMP Connection object:** CMP Connection associated with CONNTYPE_XATM_ENLIST acceptor. The definition of an [MS-CMP] connection object is extended to include the following element.
  - A reference to an XA Subordinate Enlistment object.

3.4.1.1 CONNTYPE_XATM_OPEN Acceptor States

The XA Resource Manager Bridge Facet MUST act as an acceptor for the CONNTYPE_XATM_OPEN connection type. In this role, the XA Resource Manager Bridge Facet MUST provide support for the following states.

- Idle
- Active
- Ended

3.4.1.1.1 Idle

This is the initial state. The following event is processed in this state.

- Receiving an XATMUSER_MTAG_RMOPEN Message

3.4.1.1.2 Active

The following events are processed in this state.

- Connection Disconnected
- Connection Down

3.4.1.1.3 Ended

This is the final state.

3.4.1.1.4 State Diagram

The following figure shows the relationship between the CONNTYPE_XATM_OPEN acceptor states.

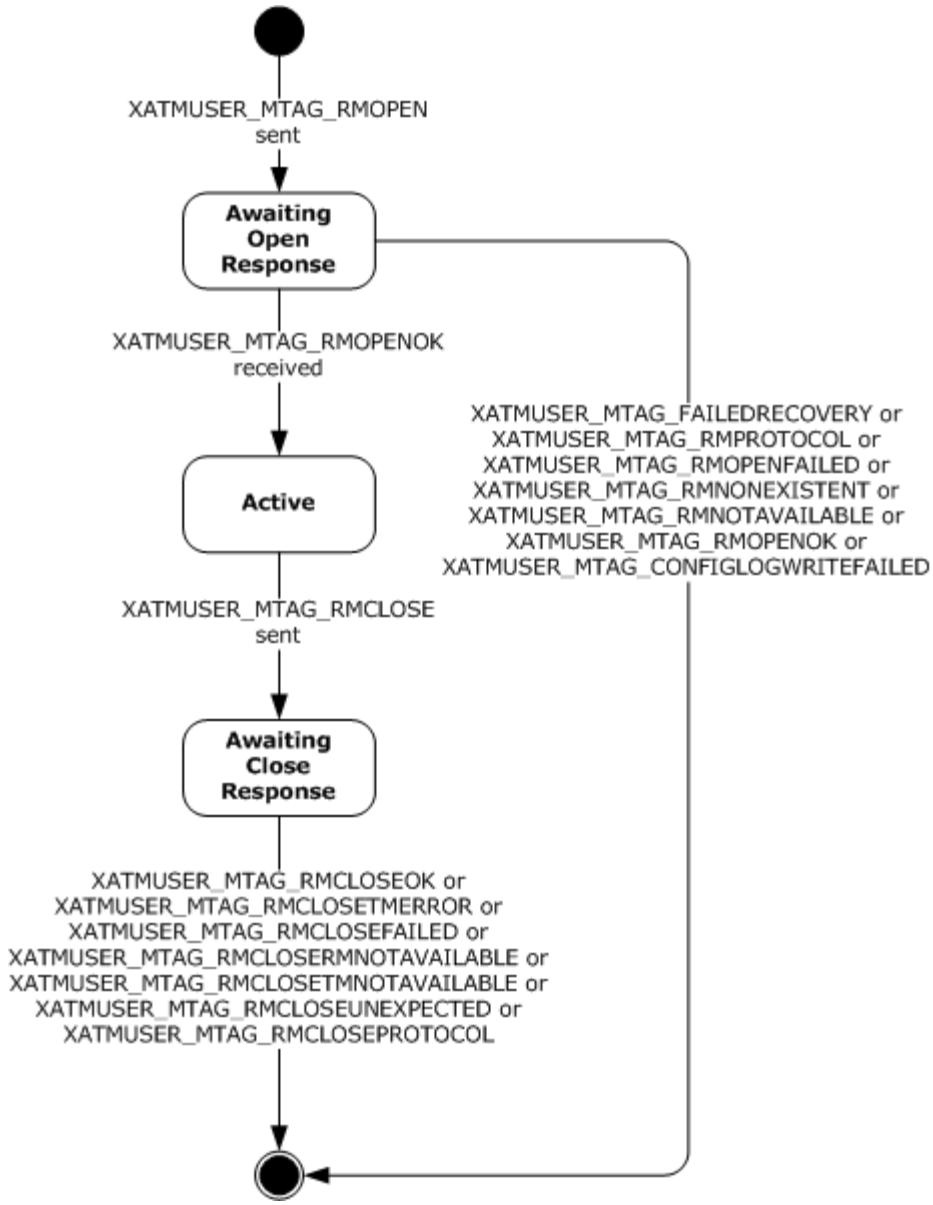


Figure 26: Relationship between the CONNTYPE_XATM_OPEN acceptor states

3.4.1.2 CONNTYPE_XATM_OPENONEPIPE Acceptor States

The XA Resource Manager Bridge Facet MUST act as an acceptor for the CONNTYPE_XATM_OPENONEPIPE connection type. In this role, the XA Resource Manager Bridge Facet MUST provide support for the following states.

- Idle
- Active
- Ended

3.4.1.2.1 Idle

This is the initial state. The following event is processed in this state.

- Receiving an XATMUSER_MTAG_RMOPEN Message

3.4.1.2.2 Active

The following events are processed in this state.

- Receiving an XATMUSER_MTAG_RMCLOSE Message
- Connection Disconnected
- Connection Down

3.4.1.2.3 Ended

This is the final state.

3.4.1.2.4 State Diagram

The following figure shows the relationship between the CONNTYPE_XATM_OPENONEPIPE acceptor states.

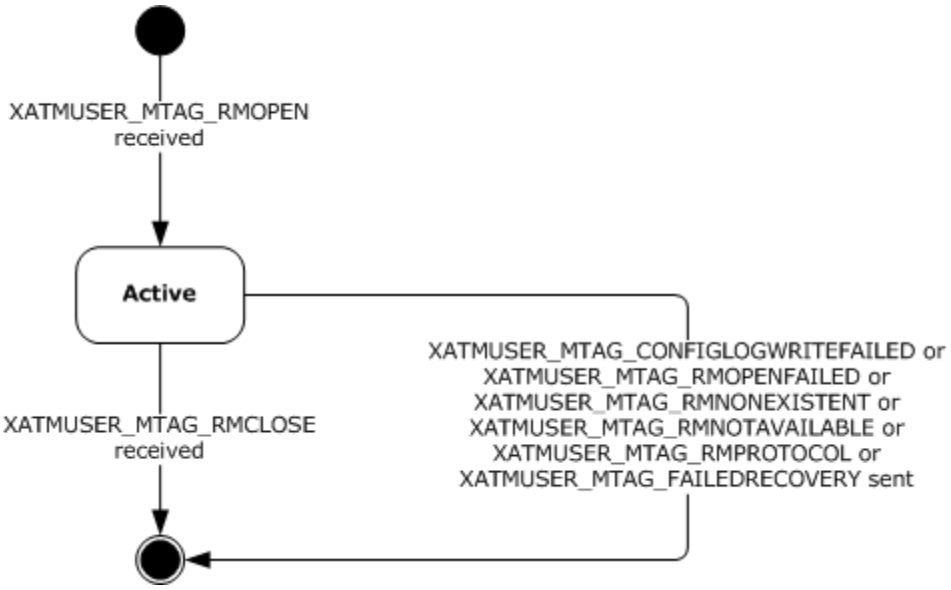


Figure 27: CONNTYPE_XATM_OPENONEPIPE acceptor states

3.4.1.3 CONNTYPE_XATM_ENLIST Acceptor States

The XA Resource Manager Bridge Facet MUST act as an acceptor for the CONNTYPE_XATM_ENLIST connection type. In this role, the XA Resource Manager Bridge Facet MUST provide support for the following states.

- Idle

- Active
- Ended

3.4.1.3.1 Idle

This is the initial state. The following event is processed in this state.

- Receiving an XATMUSER_MTAG_ENLIST Message

3.4.1.3.2 Active

The following events are processed in this state.

- Connection Disconnected
- Connection Down

3.4.1.3.3 Ended

This is the final state.

3.4.1.3.4 State Diagram

The following figure shows the relationship between the CONNTYPE_XATM_ENLIST acceptor states.

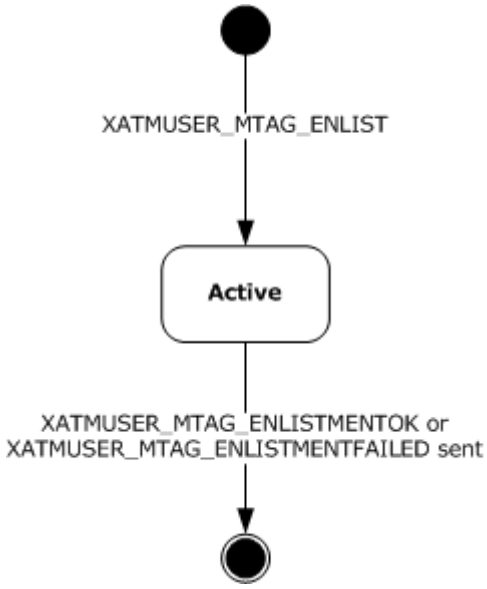


Figure 28: CONNTYPE_XATM_ENLIST acceptor states

3.4.2 Timers

The XA Resource Manager Bridge Facet MUST provide the timer specified in the next section.

3.4.2.1 Recovery Interval Timer

The default value of the timer is XATmMinWarmRecoveryInterval.

When the timer is initialized, the XA Resource Manager Bridge Facet MUST specify an XA Resource Manager object to associate with the timer. When the timer expires, the same XA Resource Manager object MUST be provided when triggering the Recovery Interval Timer event. The XA Resource Manager Bridge Facet MUST specify a distinct Recovery Interval Timer instance for each recovering XA Resource Manager object.

3.4.3 Initialization

When the XA Resource Manager Bridge Facet is initialized:

- The XA Resource Manager Bridge Facet SHOULD examine the XA Transactions Enabled flag and perform the following actions.<48>
  - If the XA Transactions Enabled flag is not set, the XA Resource Manager Bridge Facet SHOULD refuse to accept incoming connections of the following connection types.<49>
    - CONNTYPE_XATM_OPEN
    - CONNTYPE_XATM_OPENONEPIPE
    - CONNTYPE_XATM_ENLIST
- XaTmMinWarmRecoveryInterval SHOULD be set to a value that is obtained from an implementation-specific source.
- XaTmMaxWarmRecoveryInterval SHOULD be set to a value that is obtained from an implementation-specific source.

3.4.3.1 XA Resource Manager Bridge Facet Initialization

The XA Resource Manager Bridge Facet MUST perform the following initialization steps when the facet is initialized:

- XaTmMinWarmRecoveryInterval SHOULD be set to an implementation-specific value.<50>
- XaTmMaxWarmRecoveryInterval SHOULD be set to an implementation-specific value.<51>

3.4.4 Higher-Layer Triggered Events

The operation of the XA Resource Manager Bridge Facet MUST be prepared to process the higher-layer events in this section.

3.4.4.1 Recovery Event

This event is triggered by the higher-layer software hosting infrastructure when it reinitializes the system after a software failure or restart.

When the XA Resource Manager Bridge Facet is asked to recover after a software failure or restart, it MUST perform the following actions.

- For each XA Resource Manager object in the XA Resource Manager Durable Log of the XA Resource Manager Bridge Facet:
  - Attempt to create a new XA Resource Manager object with the following values.
    - The resource manager Global Identifier field is set from XA Resource Manager Durable Log.

- The resource manager Identifier field is generated by incrementing the XA Resource Manager Identifier Index field of the XA Resource Manager Bridge Facet.
- The Data Source Name field is set from XA Resource Manager Durable Log.
- The State field is set to Recovering.
- The XA Switch is set to loaded XA Switch.
- The Recovery Interval field SHOULD be set to XaTmMinWarmRecoveryInterval.<52>
- The Single Pipe field is set to FALSE.
- The Call Count field is set to 0.
- The XA DLL Name is set from the XA Resource Manager Durable Log.

- If the object is created successfully:
  - Attempt to load XA Switch for XA Resource Manager object using the XA DLL Name of the XA Resource Manager object.
  - If successful:
    - Add the new XA Resource Manager object to the XA Resource Manager Table referenced by the XA Resource Manager Bridge Facet.

- For each XA Resource Manager object in the XA Resource Manager Table of the XA Resource Manager Bridge Facet:
  - Signal Recover XA Resource Manager event on XA Resource Manager Bridge Facet with the following argument.
    - The XA Resource Manager object.

3.4.5 Message Processing Events and Sequencing Rules

3.4.5.1 CONNTYPE_XATM_OPEN as Acceptor

For all messages received in this Connection Type, the XA Resource Manager Bridge Facet MUST process the message as specified in the Common Details section. The XA Resource Manager Bridge Facet MUST also follow the processing rules specified in the following sections.

3.4.5.1.1 Receiving an XATMUSER_MTAG_RMOPEN Message

When the XA Resource Manager Bridge Facet receives an XATMUSER_ MTAG_RMOPEN message, it MUST perform the following actions.

- Test if all of the following conditions are satisfied.
  - The State field of the receiving Connection is set to Idle.
  - SHOULD check if the **lenDSN** field of the message is less than an implementation specific value.<53>
  - SHOULD check if the **lenXaDll** field of the message is less than less than an implementation-specific value.<54>

- If all the conditions are met:
  - Attempt to find an XA Resource Manager object in the XA Resource Manager Table that meets the following conditions.
    - The Data Source Name field is set to the value of the **DSN** field of the message.
  - If an XA Resource Manager object is found:
    - If the State field of the found XA Resource Manager object is set to Recovering:
      - Set the XA Resource Manager object referenced by the receiving connection to the located XA Resource Manager object.
      - Increment the Call Count field of the XA Resource Manager object.
      - Add the receiving connection to the Pending Open Connection Table referenced by the located XA Resource Manager object.
    - Otherwise, if the State field of XA Resource Manager object is set to Active:
      - Increment the Call Count field of the XA Resource Manager object.
      - Set the State field of the Connection to Active.
      - Add the receiving connection to Request Connections Table referenced by the found XA Resource Manager object.
      - Set the XA Resource Manager object of the receiving Connection to the located XA Resource Manager object.
      - Send an XATMUSER_MTAG_RMOPENOK on the receiving connection with the following arguments.
        - Resource Manager Identifier field of the located XA Resource Manager object.
        - Resource Manager Global Identifier field of the located XA Resource Manager object.
    - Otherwise:
      - Send an XATMUSER_MTAG_E_RMNOTAVAILABLE message on the receiving connection.
  - Otherwise:
    - Attempt to load XA Switch for XA Resource Manager using the **XaDllFileName** field of the message:
    - If successful:
      - Attempt to create a new XA Resource Manager object with the following values.
        - The resource manager Global Identifier field is set to newly created GUID.
        - The resource manager Identifier field is set to incremented XA Resource Manager Identifier Index.
        - The Data Source Name field is set to the **DSN** field of the message.
        - The State field is set to Idle.

- The XA Switch set to loaded XA Switch.
- The Recovery Interval field SHOULD be set to XaTmMinWarmRecoveryInterval.<55>
- The Single Pipe field set to FALSE.
- The Call Count field is set to 1.
- The XA DLL Name field is set to the **XaDllFileName** field of the message.

- If successful:
  - Set the XA Resource Manager of the receiving connection to the created XA Resource Manager object.
  - Call xa_open() on the XA Switch referenced by the new XA Resource Manager object.
  - If the outcome is set to XA_OK:
    - Otherwise:
      - Write the created XA Resource Manager object to the XA Resource Manager Durable Log.
      - Add the created XA Resource Manager object to the Active Resource Manager Table referenced by the Core Transaction Manager Facet keyed by the resource manager Global Identifier field.
      - Set the State field of the XA Resource Manager object to Active.
      - Add receiving connection to Request Connections Table referenced by the XA Resource Manager object.
      - Set the State of Connection to Active.
      - Send an XATMUSER_MTAG_RMOPENOK on the receiving connection with the following arguments.
        - The resource manager Identifier field of the created XA Resource Manager.
        - The resource manager Global Identifier field of the created XA Resource Manager.
  - Otherwise if outcome is set to XAER_PROTO:
    - Set the State field of the created XA Resource Manager object to Ended.
    - Send an XATMUSER_MTAG_E_RMPROTOCOL message on the receiving connection.
    - Set the Connection State to Ended.
  - Otherwise:<56>
    - Set the State field of the created XA Resource Manager object to Ended.
    - Send an XATMUSER_MTAG_E_RMOPENFAILED on the receiving connection.
    - Set Connection State to Ended.

- Otherwise:
  - Send an XATMUSER_MTAG_E_RMOPENFAILED message using the receiving connection.
  - Set Connection State to Ended.
- Otherwise:
  - Send an XATMUSER_MTAG_E_RMOPENFAILED message using the receiving connection.

    Set Connection State to Ended.

- Otherwise:
  - If second or third condition failed, XA Resource Manager Bridge Facet MAY:<57>
    - Send an XATMUSER_MTAG_E_RMOPENFAILED message using the receiving connection.
    - Set the Connection State to Ended.

3.4.5.1.2 Connection Disconnected, Connection Down

When a CONNTYPE_ XATM_OPEN connection is disconnected, the XA Resource Manager Bridge Facet MUST perform the following actions.

- If the State of receiving Connection is set to Active and the State field of the XA Resource Manager object referenced by the XA Resource Manager CMP Connection is set to Active:
  - Set the State of receiving Connection to Ended.
  - Decrement the Call Count field of the XA Resource Manager object referenced by the XA Resource Manager CMP Connection.
  - Remove the receiving connection from Request Connections Table referenced by XA Resource Manager object referenced by the XA Resource Manager CMP Connection.
  - If the Call Count field of the XA Resource Manager object referenced by the XA Resource Manager CMP Connection is set to 0:
    - Call xa_close() on XA Switch of XA Resource Manager object with the following arguments.
      - The Data Source Name field of the XA Resource Manager object referenced by the XA Resource Manager CMP Connection.
      - The resource manager Global Identifier field of the XA Resource Manager object referenced by the XA Resource Manager CMP Connection.
      - TMNOFLAGS.
    - Set the State field of the XA Resource Manager object referenced by the XA Resource Manager CMP Connection to Ended.
    - If the XA Subordinate Enlistment Table referenced by the XA Resource Manager object referenced by the XA Resource Manager CMP Connection is empty:
      - Remove the XA Resource Manager object referenced by the XA Resource Manager CMP Connection from the XA Resource Manager Durable Log.

- Remove the XA Resource Manager referenced by the XA Resource Manager CMP Connection from the Active Resource Manager Table referenced by the transaction manager.
- Remove the XA Resource Manager referenced by the XA Resource Manager CMP Connection from the XA Resource Manager Table.

3.4.5.2 CONNTYPE_XATM_OPENONEPIPE as Acceptor

For all messages received in this Connection Type, the XA Resource Manager Bridge Facet MUST process the message as specified in the Common Details section. The XA Resource Manager Bridge Facet MUST also follow the processing rules specified in the following sections.

3.4.5.2.1 Receiving an XATMUSER_MTAG_RMOPEN Message

When the XA Resource Manager Bridge Facet receives an XATMUSER_ MTAG_RMOPEN message, it MUST perform the following actions.

- Test if all of the following conditions are satisfied.
  - The State field of the receiving Connection is set to Idle.
  - SHOULD check if the **lenDSN** field of the message is less than an implementation specific value.<58>
  - SHOULD check if the **lenXaDll** field of the message is less than less than an implementation specific value.<59>
- If all the conditions are met:
  - Attempt to load XA Switch for XA Resource Manager using the **XaDllFileName** field of the message.
  - If successful:
    - Randomly generate a Resource Manager Identifier.
    - Call xa_open() on the XA Switch of the XA Resource Manager object with the following arguments.
      - **DSN** field of the message.
      - Previously generated Resource Manager Identifier.
      - TMNOFLAGS.
    - If the outcome of the call to xa_open() is set to XA_OK:
      - Call xa_close() on the XA Switch with the following arguments.
        - **DSN** field of the message
        - Previously generated Resource Manager Identifier
        - TMNOFLAGS
      - If the outcome of the call to xa_close() is set to XA_OK:

- Attempt to create a new XA Resource Manager object with the following values.
  - The resource manager Global Identifier field is set to newly created GUID.
  - The Randomly generated Resource Manager Identifier.
  - The Data Source Name field is set to the **DSN** field of the message.
  - The State field is set to Idle.
  - The XA Switch is set to loaded XA Switch.
  - The Recovery Interval field SHOULD be set to XaTmMinWarmRecoveryInterval.<60>
  - The Single Pipe field is set to TRUE.
  - The Call Count field is set to 1.
  - The XA DLL Name field is set to the **XaDllFileName** field of the message.
- If successful:
  - Set the XA Resource Manager of the receiving connection to the new XA Resource Manager.
  - Write the new XA Resource Manager to the XA Resource Manager Durable Log.
  - Set the State of Connection to Active.
  - Send an XATMUSER_MTAG_RMOPENOK message on the receiving connection with the following arguments.
    - Resource Manager Identifier of the created XA Resource Manager.
    - Resource Manager Global Identifier of the created XA Resource Manager.
- Otherwise:
  - Send an XATMUSER_MTAG_E_RMOPENFAILED message on the receiving connection.
- Otherwise:
  - Send an XATMUSER_MTAG_E_RMOPENFAILED message on the receiving connection.

- Otherwise the message MUST be processed as an invalid message as specified in section 3.1.5.

3.4.5.2.2 Receiving an XATMUSER_MTAG_RMCLOSE Message

When the XA Resource Manager Bridge Facet receives a XATMUSER_ MTAG_RMCLOSE message, it MUST perform the following actions.

- If the State field of receiving Connection is set to Active:
  - Attempt to delete the XA Resource Manager object referenced by the XA Resource Manager CMP Connection from the XA Resource Manager Durable Log.
  - If the delete is successful:

- Send an XATMUSER_MTAG_RMCLOSEOK message using the receiving connection.

- Otherwise:

  - Send XATMUSER_MTAG_E_RMCLOSEFAILED to all connections in Request Connections Table of XA Resource Manager.

- Set State of receiving connection to Ended.

3.4.5.2.3 Connection Disconnected, Connection Down

When a CONNTYPE_ XATM_OPENONEPIPE connection is disconnected, the XA Resource Manager Bridge Facet MUST perform the following actions.

- If State field of the receiving Connection is set to Active:

  - Trigger the Recover XA Resource Manager event on XA Resource Manager Bridge Facet with the following argument.

    - The XA Resource Manager object referenced by the XA Resource Manager CMP Connection.

3.4.5.3 CONNTYPE_XATM_ENLIST as Acceptor

For all messages received in this Connection Type, the XA Resource Manager Bridge Facet MUST process the message as specified in the Common Details section. The XA Resource Manager Bridge Facet MUST also follow the processing rules specified in the following sections.

3.4.5.3.1 Receiving an XATMUSER_MTAG_ENLIST Message

When the XA Resource Manager Bridge Facet receives a XATMUSER_ MTAG_ENLIST message, it MUST perform the following actions.

- Test if all of the following conditions are satisfied.

  - The State field of receiving Connection is set to Idle.

- If all the conditions are met:

  - Attempt to find an XA Resource Manager object in the XA Resource Manager Table that meets the following condition.

    - The resource manager Global Identifier field is set to **guidRM** field of the message.

  - If condition is met:

    - If the State field of the XA Resource Manager object is set to Active:

      - Attempt to find an XA Subordinate Enlistment object in the XA Subordinate Enlistment Table referenced by the found XA Resource Manager object that meets the following condition.

        - The GTRID field of the XID structure of the XA Subordinate Enlistment object is set to the GTRID field of the Xid field of the message.

      - If an object is found:

        - Set the State field of the receiving Connection to Ended.

- Send an XATMUSER_MTAG_E_ENLISTMENTDUPLICATE message on the receiving Connection.

- Otherwise:
  - Attempt to create an XA Subordinate Enlistment object with the following values.
    - The State field is set to Idle.
    - The resource manager Identifier field is set to **guidRM** field of the message.
    - The XID field is set to the Xid field of the message.
    - The Current Request Connection field is set to the receiving Connection.
  - If successful:
    - Add the newly created XA Subordinate Enlistment object to the XA Subordinate Enlistment Table referenced by the found XA Resource Manager object.
    - Attempt to Import a transaction as specified in section 3.3.4.5 of [MS-DTCO] with the following value.
      - The ImportCookie of the received message.
    - If successful:
      - Set the State field of the receiving connection to Active.
      - Set the XA Subordinate Enlistment object referenced by the receiving connection to the created XA Subordinate Enlistment object.
      - Set the Transaction referenced by the created XA Subordinate Enlistment object to the Imported Transaction.
      - Signal the Create Subordinate Enlistment event on the Core Transaction Manager Facet with the following argument.
      - The created XA Subordinate Enlistment object.
    - Otherwise:
      - Remove the created XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by the found XA Resource Manager object.
      - Set the State of receiving connection to Ended.
      - Send an XATMUSER_MTAG_E_ENLISTMENTIMPFAILED using the receiving connection.
  - Otherwise:
    - Set the State of receiving connection to Ended.
    - Send an XATMUSER_MTAG_E_ENLISTMENTNOMEMORY message using the receiving connection.
- Otherwise, if the State field of the XA Resource Manager object is set to Ended:

- Set the State of receiving connection to Ended.
- Send an XATMUSER_MTAG_E_ENLISTMENTTOOLATE message using the receiving connection.

- Otherwise:
  - Set the State of the receiving Connection to Ended.
  - Send an XATMUSER_MTAG_E_ENLISTMENTRMRECOVERING message using the receiving connection.

- Otherwise:
  - Set the State of receiving Connection to Ended.
  - Send an XATMUSER_MTAG_E_ENLISTMENTRMNOTFOUND message using the receiving connection.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.4.5.3.2 Connection Disconnected, Connection Down

When a CONNTYPE_ XATM_ENLIST connection is disconnected, the XA Resource Manager Bridge Facet MUST perform the following actions.

- If the State of receiving Connection is set to Active:
  - Set the State of receiving Connection to Ended.
  - Reset the Current Request Connection field of the XA Subordinate Enlistment object referenced by the receiving connection.

3.4.6 Timer Events

3.4.6.1 Recovery Interval Timer

When this timer expires, the XA Resource Manager Bridge Facet MUST perform the following actions.

- Signal the Recover XA Resource Manager event on the XA Resource Manager Bridge Facet with the following argument.
  - The provided XA Resource Manager object.

3.4.7 Other Local Events

3.4.7.1 Begin Commit

The Begin Commit event MUST be signaled with the following arguments.

- An XA Subordinate Enlistment object.

If the Begin Commit event is signaled, the XA Resource Manager Bridge Facet MUST perform the following actions.

- If the State field of the provided XA Subordinate Enlistment object is set to Prepared:

- Call xa_commit() on the XA Switch of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object with the following arguments.
  - The XID field of the provided XA Subordinate Enlistment object.
  - The resource manager Identifier field of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
  - TMNOFLAGS.
- If the Output is either XAER_RMFAIL, XA_RETRY, XAER_RMERR, XAER_NOTA, XAER_INVAL, or XAER_PROTO:
  - Set the State field of the provided XA Subordinate Enlistment object to Ended.
  - Remove the provided XA Subordinate Enlistment object from XA Subordinate Enlistment Table referenced by XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
  - Signal the Recover XA Resource Manager event on the XA Resource Manager Bridge Facet with the following argument.
  - The XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
- Otherwise:
  - Set the State field of the XA Subordinate Enlistment object to Ended.
  - Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by XA Resource Manager object referenced by provided XA Subordinate Enlistment object.
  - Signal the Enlistment Commit Complete event on the Core Transaction Manager Facet with the following arguments.
    - The provided XA Subordinate Enlistment object.
- If the XA Subordinate Enlistment Table of XA Resource Manager referenced by the provided XA Subordinate Enlistment object is empty and the State field of the XA Resource Manager object is set to Ended:
  - Remove the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object from the XA Resource Manager Durable Log.
  - Remove the XA Resource Manager referenced by the provided XA Subordinate Enlistment object from the transaction manager's Active Resource Manager Table.
  - Remove the XA Resource Manager referenced by the provided XA Subordinate Enlistment object from the XA Resource Manager Table.

3.4.7.2 Begin Phase One

The Begin Phase One event MUST be signaled with the following arguments.

- An XA Subordinate Enlistment object.
- Single Phase flag.

If the Begin Phase One event is signaled, the XA Resource Manager Bridge Facet MUST perform the following actions.

- If the State field of the provided XA Subordinate Enlistment object is set to Enlisted:
  - If the provided Single Phase flag is set to FALSE:
    - Call xa_prepare() on the XA Switch of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object with the following arguments.
      - The XID field of the provided XA Subordinate Enlistment object.
      - The resource manager Identifier field of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
      - TMNOFLAGS.
    - If the Output is set to XA_OK:
      - Set the State field of the XA Subordinate Enlistment object to Prepared.
      - Signal the Enlistment Phase One Complete event on the Core Transaction Manager Facet with the following arguments.
        - The provided XA Subordinate Enlistment object.
        - The Outcome set to Prepared.
    - If the Output is set to XA_RDONLY:
      - Signal the Enlistment Phase One event on Core Transaction Manager Facet with the following arguments.
        - The provided XA Subordinate Enlistment object.
        - The Outcome set to Read Only.
      - Set the State field of the XA Subordinate Enlistment object to Ended.
      - Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
    - If the Output is set to XAER_RMFAIL or XAER_RMERR:
      - Signal the Enlistment Phase One Complete event on the Core Transaction Manager Facet with the following arguments.
        - The provided XA Subordinate Enlistment object.
        - The Outcome set to Aborted.
      - Set the State of the provided XA Subordinate Enlistment object to Ended.
      - Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.

- Signal the Recover XA Resource Manager event on the XA Resource Manager Bridge Facet with the following argument.
  - The XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
- Otherwise:
  - Signal the Enlistment Phase One Complete event on the Core Transaction Manager Facet with the following arguments.
    - The provided XA Subordinate Enlistment object.
    - The Outcome set to Aborted.
  - Set the State field of the provided XA Subordinate Enlistment object to Ended.
  - Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by XA Resource Manager object referenced by the provided XA Subordinate Enlistment.
- Otherwise:
  - Call xa_commit() on the XA Switch of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object with the following arguments.
    - The XID field of the provided XA Subordinate Enlistment object.
    - The resource manager Identifier field of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
    - TMONEPHASE.
  - If the Output is set to XA_OK:
    - Signal the Enlistment Phase One Complete event on the Core Transaction Manager Facet with the following arguments.
      - The provided XA Subordinate Enlistment object.
      - The Outcome set to Committed.
    - Set the State field of the XA Subordinate Enlistment object to Ended.
    - Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by XA Resource Manager object referenced by provided XA Subordinate Enlistment object.
  - If the Output is set to XAER_RMFAIL:
    - Signal Enlistment Phase One Complete event on the Core Transaction Manager Facet with the following arguments.
      - The provided XA Subordinate Enlistment object.
      - The Outcome set to Aborted.
    - Set the State field of the XA Subordinate Enlistment object to Ended.

- Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by XA Resource Manager object referenced by provided XA Subordinate Enlistment.
- Signal the Recover XA Resource Manager event on XA Resource Manager Bridge Facet with the following argument.
  - The XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.

- Otherwise:
  - Signal Enlistment Phase One Complete event on the Core Transaction Manager Facet with the following arguments.
    - The provided XA Subordinate Enlistment object.
    - The Outcome set to Aborted.
  - Set the State field of the provided XA Subordinate Enlistment object to Ended.
  - Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by the XA Resource Manager referenced by the provided XA Subordinate Enlistment object.

- If the XA Subordinate Enlistment Table of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object is empty and the State field of XA Resource Manager referenced by the provided XA Superior Enlistment object is set to Ended:
  - Remove the XA Resource Manager referenced by the provided XA Subordinate Enlistment object from the XA Resource Manager Durable Log.
  - Remove the XA Resource Manager referenced by the provided XA Subordinate Enlistment object from the transaction manager's Active Resource Manager Table.
  - Remove the XA Resource Manager referenced by the provided XA Subordinate Enlistment object from the XA Resource Manager Table.

3.4.7.3 Begin Rollback

The Begin Rollback event MUST be signaled with the following arguments.

- An XA Subordinate Enlistment object.

If the Begin Rollback event is signaled, the XA Resource Manager Bridge Facet MUST perform the following actions.

- If the State field of the provided XA Subordinate Enlistment object is set to Enlisted or Prepared:
  - Call xa_rollback() on the XA Switch of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object with the following arguments.
    - The XID field of the provided XA Subordinate Enlistment object.
    - The resource manager Identifier field of the XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
    - TMNOFLAGS.

- If the Output is set to either XAER_RMFAIL, XA_RETRY, XAER_RMERR, XAER_NOTA, XAER_INVAL, or XAER_PROTO:
  - Set the State field of the XA Subordinate Enlistment object to Ended.
  - Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by the XA Resource Manager object referenced by provided XA Subordinate Enlistment object.
  - Signal the Recover XA Resource Manager event on XA Resource Manager Bridge Facet with the following argument.
    - The XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
- Otherwise:
  - Set the State field of the XA Subordinate Enlistment object to Ended.
  - Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by the XA Resource Manager referenced by the provided XA Subordinate Enlistment.
  - Signal the Enlistment Rollback Complete event on the Core Transaction Manager Facet with the following argument.
    - The Provided XA Subordinate Enlistment object.
- If the XA Subordinate Enlistment Table referenced by the XA Resource Manager referenced by the provided XA Subordinate Enlistment object is empty and State of XA Resource Manager referenced by the provided XA Subordinate Enlistment object is set to Ended:
  - Remove the XA Resource Manager referenced by the provided XA Subordinate Enlistment object from the XA Resource Manager Durable Log.
  - Remove the XA Resource Manager referenced by the provided XA Subordinate Enlistment object from the transaction manager's Active Resource Manager Table.
  - Remove the XA Resource Manager referenced by the provided XA Subordinate Enlistment object from the XA Resource Manager Table.

3.4.7.4 Create Subordinate Enlistment Failure

The Create Subordinate Enlistment Failure event MUST be signaled with the following arguments.

- An XA Subordinate Enlistment object.

If the Create Subordinate Enlistment Failure event is signaled, the XA Resource Manager Bridge Facet MUST perform the following actions.

- Remove the provided XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by the XA Resource Manager referenced by the provided XA Subordinate Enlistment object.
- Set the State field of the provided XA Subordinate Enlistment object to Ended.
- Send an XATMUSER_MTAG_E_ENLISTMENTFAILED message using the Current Request Connection field of the provided XA Subordinate Enlistment object.

- Set the State of the Current Request Connection field of the provided XA Subordinate Enlistment object to Ended.

3.4.7.5 Create Subordinate Enlistment Success

The Create Subordinate Enlistment Success event MUST be signaled with the following arguments.

- An XA Subordinate Enlistment object.

If the Create Subordinate Enlistment Success event is signaled, the XA Resource Manager Bridge Facet MUST perform the following actions.

- Set the State field of the provided XA Subordinate Enlistment object to Enlisted.
- Send an XATMUSER_MTAG_ENLISTMENTOK message using the Current Request Connection field of the provided XA Subordinate Enlistment object.
- Set the State of the Current Request Connection field of the provided XA Subordinate Enlistment object to Ended.

3.4.7.6 Recover XA Resource Manager

The Recover XA Resource Manager event MUST be signaled with the following arguments.

- An XA Resource Manager object.

If the Recover XA Resource Manager event is signaled, the XA Resource Manager Bridge Facet MUST perform the following actions.

- Set the State field of the provided XA Resource Manager object to Recovering.
- Create a local Recovery Flags variable that is set to TMSTARTSCAN.
- For each XA Subordinate Enlistment object in the XA Subordinate Enlistment Table referenced by provided XA Resource Manager object:
  - Set the State field of the XA Subordinate Enlistment object to Ended.
  - Remove the XA Subordinate Enlistment object from the XA Subordinate Enlistment Table referenced by the provided XA Resource Manager object.
- Call xa_open() on the XA Switch referenced by the XA Resource Manager object with the following arguments.
  - The Data Store Name field of the provided XA Resource Manager object.
  - The resource manager Identifier field of the provided XA Resource Manager object.
  - TMNOFLAGS.
- If the result is set to XA_OK:
  - Repeat the following steps until the return value from xa_recover() is less than 10:
    - Call xa_recover() on the XA Switch of the provided XA Resource Manager with the following arguments.
      - XA_UOW xidbuffer[10].

        - 10.
        - Resource Manager Identifier field of XA Resource Manager object referenced by the provided XA Subordinate Enlistment object.
        - Recovery Flags.
    - Set Recovery Flags to TMNOFLAGS.
    - Repeat the following steps for each of the XIDs passed back in xidbuffer:
        - Test if the following set of conditions are met:
            - XID from xidbuffer has XA_BQUAL_1 with XATMGUID set to XA Transaction Manager GUID.
            - XID from xidbuffer has XA_BQUAL_1 with RMguid set to Resource Manager Global Identifier of XA Resource Manager object.
        - If the conditions are satisfied.
            - Attempt to find a transaction in the Transaction Table referenced by the Core Transaction Manager Facet which has Transaction Identifier equal to the GTRID field of the XID structure from xidbuffer.
            - If found:
                - If the State of the found Transaction is set to Committed:
                    - Call xa_commit() on the XA Switch of the provided XA Resource Manager object with the following arguments.
                        - The XID from xidbuffer.
                        - The resource manager Identifier field of the provided XA Resource Manager.
                        - TMNOFLAGS.
                    - If the Outcome is set to XA_RETRY, XAER_RMFAIL, or XAER_RMERR:
                        - Initialize a Recovery Interval Timer with a time out of the Recovery Interval field of the provided XA Resource Manager object and the following argument.
                            - The provided XA Resource Manager object.
                        - The Recovery Interval field of the provided XA Resource Manager object SHOULD be set to double its previous value, with a ceiling of XaTmMaxWarmRecoveryInterval.<61>
                    - Otherwise, if the Outcome is set to either XA_HEURCOM or XA_OK:
                        - Continue to next XID to be recovered.
                    - Otherwise:
                        - Set the State field of the XA Resource Manager object to Ended.

- Remove the provided XA Resource Manager object from the XA Resource Manager Durable Log.
- Remove the provided XA Resource Manager object from the XA Resource Manager Table.
- If Pending Open Connection Table referenced by the XA Resource Manager object is not empty:
  - The following actions MUST be performed on each XA Resource Manager CMP Connection in the Pending Open Connection Table referenced by the XA Resource Manager object.

Remove XA Resource Manager CMP Connection from the Pending Open Connection Table referenced by the XA Resource Manager object.

Set State field of XA Resource Manager CMP Connection to Ended.

If the Outcome of the xa_rollback() call was XAER_PROTO send an XATMUSER_MTAG_E_RMPROTOCOL message on the XA Resource Manager CMP Connection.

Otherwise send an XATMUSER_MTAG_E_RMOPENFAILED message on the XA Resource Manager CMP Connection.

- Terminate the processing for this event.

- Otherwise:
  - Call xa_rollback() on the XA Switch of the provided XA Resource Manager object with the following arguments.
    - The XID from xidbuffer.
    - The resource manager Identifier field of the provided XA Resource Manager object.
    - TMNOFLAGS.
  - If the Outcome is set to XA_RETRY, XAER_RMFAIL, or XAER_RMERR:
    - Initialize a Recovery Interval Timer with a time out of the Recovery Interval field of the provided XA Resource Manager object and the following argument.
      - The provided XA Resource Manager object.
    - The Recovery Interval field of the provided XA Resource Manager SHOULD be set to double its previous value, with a ceiling of XaTmMaxWarmRecoveryInterval.<62>
  - Otherwise, if the Outcome is set to either XA_HEURRB, XA_OK, or XA_RB*:
    - Continue to next XID to be recovered.
  - Otherwise:
    - Set the State field of the XA Resource Manager object to Ended.

- Remove the provided XA Resource Manager object from the XA Resource Manager Durable Log.
- Remove the provided XA Resource Manager object from the XA Resource Manager Table.
- If Pending Open Connection Table referenced by the XA Resource Manager object is not empty:
  - The following actions MUST be performed on each XA Resource Manager CMP Connection in the Pending Open Connection Table referenced by the XA Resource Manager object.

Remove XA Resource Manager CMP Connection from the Pending Open Connection Table referenced by the XA Resource Manager object.

Set State field of XA Resource Manager CMP Connection to Ended.

If the Outcome of the xa_rollback() call was XAER_PROTO send an XATMUSER_MTAG_E_RMPROTOCOL message on the XA Resource Manager CMP Connection.

Otherwise send an XATMUSER_MTAG_E_RMOPENFAILED message on the XA Resource Manager CMP Connection.

- Terminate the processing for this event.

- Otherwise:
  - Call xa_rollback() on the XA Switch of the provided XA Resource Manager object with the following arguments.
    - The XID from xidbuffer.
    - The resource manager Identifier field of the provided XA Resource Manager object.
    - TMNOFLAGS.
  - If the Outcome is set to XA_RETRY, XAER_RMERR, or XAER_RMFAIL:
    - Initialize a Recovery Interval Timer with a time out of the Recovery Interval field of the provided XA Resource Manager object and the following argument.
      - The provided XA Resource Manager object.
    - The Recovery Interval field of the provided XA Resource Manager SHOULD be set to double its previous value, with a ceiling of XaTmMaxWarmRecoveryInterval.<63>
  - Otherwise, if the Outcome is set to either XA_HEURRB, XA_OK, or XA_RB*:
    - Continue to next XID to be recovered.
  - Otherwise:
    - Set the State field of the XA Resource Manager object to Ended.
    - Remove the provided XA Resource Manager object from the XA Resource Manager Durable Log.

- Remove the provided XA Resource Manager object from the XA Resource Manager Table.
- If Pending Open Connection Table referenced by the XA Resource Manager object is not empty:
  - The following actions MUST be performed on each XA Resource Manager CMP Connection in the Pending Open Connection Table referenced by the XA Resource Manager object.
    - Remove XA Resource Manager CMP Connection from the Pending Open Connection Table referenced by the XA Resource Manager object.
    - Set State field of XA Resource Manager CMP Connection to Ended.
    - If the Outcome of the xa_rollback() call was XAER_PROTO:

Send an XATMUSER_MTAG_E_RMPROTOCOL message on the XA Resource Manager CMP Connection.

    - Otherwise:

Send an XATMUSER_MTAG_E_RMOPENFAILED message on the XA Resource Manager CMP Connection.

  - Terminate the processing for this event.

- Call xa_close() on the XA switch of the provided XA Resource Manager object with the following arguments.
  - The Data Store Name of provided XA Resource Manager object.
  - The resource manager Identifier of XA Resource Manager object.
  - TMNOFLAGS.
- If the Open Count field of the XA Resource Manager object is set to 0:
  - Remove the provided XA Resource Manager object from the XA Resource Manager Durable Log.
  - Remove the provided XA Resource Manager object from the XA Resource Manager Table.
  - Set the State field of the XA Resource Manager object to Ended.
- Otherwise:
  - Set the State field of the XA Resource Manager object to Active.
  - If Pending Open Connection Table referenced by the XA Resource Manager object is not empty:
    - The following actions MUST be performed on each XA Resource Manager CMP Connection in the Pending Open Connection Table referenced by the XA Resource Manager object.
      - Remove XA Resource Manager CMP Connection from the Pending Open Connection Table referenced by the XA Resource Manager object.

- Add XA Resource Manager CMP Connection to the Request Connections table referenced by the XA Resource Manager.
- Set the State field of the Connection to Active.
- Send an XATMUSER_MTAG_RMOPENOK on the receiving connection with the following arguments.
  - Resource Manager Identifier field of the XA Resource Manager object.
  - Resource Manager Global Identifier field of the XA Resource Manager object.

- Otherwise if Outcome is XAER_RMFAIL, XA_RETRY, or XAER_RMERR:
  - Signal Recovery Interval Timer event with the following value.
    - The Recovery Interval field of the provided XA Resource Manager object.
    - The provided XA Resource Manager object.
  - Recovery Interval field of the provided XA Resource Manager SHOULD be set to double its previous value, with a ceiling of XaTmMaxWarmRecoveryInterval.<64>
- Otherwise:
  - Remove the provided XA Resource Manager object from the XA Resource Manager Durable Log.
  - Remove the provided XA Resource Manager object from the XA Resource Manager Table.
  - If Pending Open Connection Table referenced by the XA Resource Manager object is not empty:
    - The following actions MUST be performed on each XA Resource Manager CMP Connection in the Pending Open Connection Table referenced by the XA Resource Manager object.
      - Remove XA Resource Manager CMP Connection from the Pending Open Connection Table referenced by the XA Resource Manager object.
      - Set State field of XA Resource Manager CMP Connection to Ended.
      - If the Outcome of the xa_open() call was XAER_PROTO:
        - Send an XATMUSER_MTAG_E_RMPROTOCOL message on the XA Resource Manager CMP Connection.
      - Otherwise:
        - Send an XATMUSER_MTAG_E_RMOPENFAILED message on the XA Resource Manager CMP Connection.

3.5 XA Resource Manager Bridge Details

3.5.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate the implementations

adhere to this model as long as their external behavior is consistent with the behavior that is described in this document.

Note that the abstract data model can be implemented in a variety of ways. This protocol does not prescribe or advocate any specific implementation technique.

The XA Resource Manager Bridge MUST maintain all the data elements that are specified in section 3.1.1.

The XA Resource Manager Bridge MUST also maintain the following data elements.

- **One-Pipe XA Resource Manager Proxy Table**: A table of the currently active XA Resource Manager Proxy objects keyed by the resource manager Cookie.
- **Two-Pipe XA Resource Manager Proxy Table:** A table of the currently active XA Resource Manager Proxy objects keyed by the resource manager Cookie.
- **XA Resource Manager Proxy object:** Represents a currently active XA Resource Manager.
- **Resource Manager Global Identifier:** Specifies a unique identifier associated with an XA Resource Manager.
- **Request Connection:** Specifies the XA Resource Manager Proxy Connection to the XA Resource Manager Bridge Facet.
- **DLL Name:** Specifies the DLL name of an XA Resource Manager.
- **Data Store Name:** Implementation specific ASCII string to be passed to xa_open and xa_close calls on the XA Switch.
- **Pipe:** Identifies whether the XA Resource Manager Proxy is using the one pipe or Two Pipe protocol.
  - **One:** Uses the One-Pipe protocol.
  - **Two:** Uses the Two-Pipe protocol.
- Resource Manager: Corresponds to the resource manager described in [MS-DTCO] section 3.5.
- **XA Resource Manager Proxy CMP Connection object:** CMP Connection object extended to include the following:
  - Reference to an XA Resource Manager Proxy object.

3.5.1.1 CONNTYPE_XATM_OPEN Initiator States

The XA Resource Manager Bridge MUST act as an initiator for the CONNTYPE_XATM_OPEN connection type. In this role, the XA Resource Manager Bridge MUST provide support for the following states.

- Idle
- Awaiting Open Response
- Active
- Ended

The following figure shows the relationship between the CONNTYPE_XATM_OPEN initiator states.

3.5.1.1.1 Idle

This is the initial state.

3.5.1.1.2 Awaiting Open Response

The following events are processed in this state.

- Receiving an XATMUSER_MTAG_RMOPENOK Message
- Receiving Other XATMUSER_MTAG_RMOPEN Messages
- Connection Disconnected

3.5.1.1.3 Active

The following events are processed in this state.

- Connection Disconnected

3.5.1.1.4 Ended

This is the final state.

3.5.1.1.5 State Diagram

The following figure shows the relationship between the CONNTYPE_XATM_OPEN initiator states.

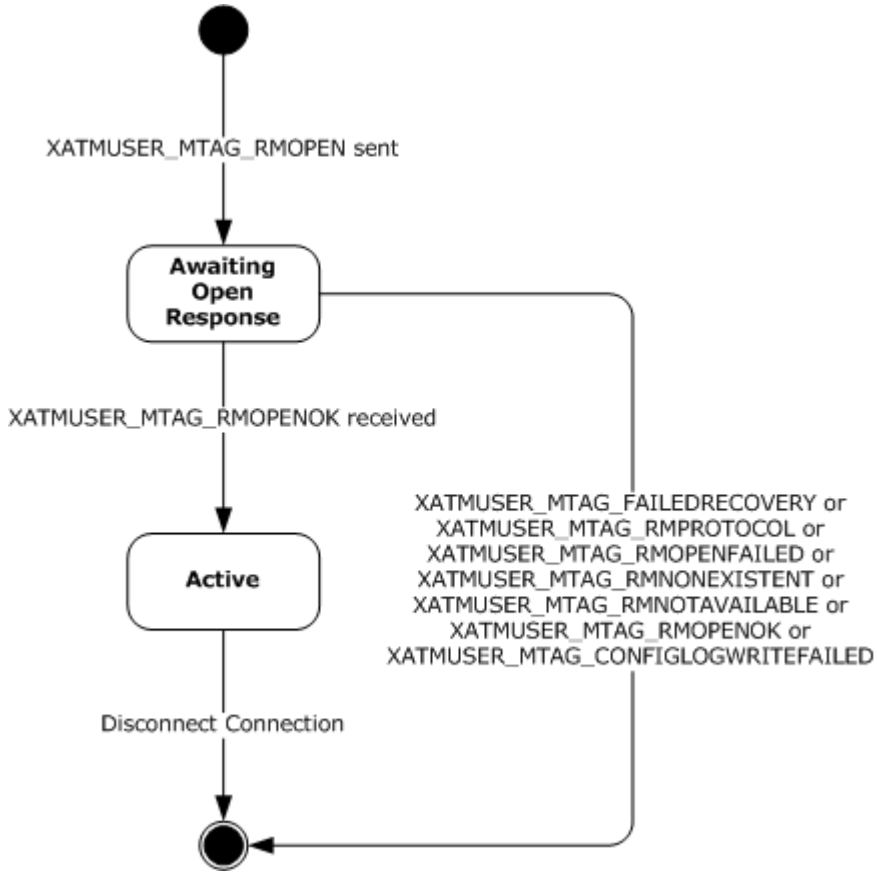


Figure 29: CONNTYPE_XATM_OPEN initiator states

3.5.1.2 CONNTYPE_XATM_OPENONEPIPE Initiator States

The XA Resource Manager Bridge MUST act as an initiator for the CONNTYPE_XATM_OPENONEPIPE connection type. In this role, the XA Resource Manager Bridge MUST provide support for the following states.

- Idle
- Awaiting Open Response
- Active
- Awaiting Close Response
- Ended

The following figure shows the relationship between the CONNTYPE_XATM_OPENONEPIPE initiator states.

3.5.1.2.1 Idle

This is the initial state. No events are processed in this state.

3.5.1.2.2 Awaiting Open Response

The following events are processed in this state.

- Receiving an XATMUSER_MTAG_RMOPENOK Message
- Receiving Other XATMUSER_MTAG_RMOPEN Messages
- Connection Disconnected

3.5.1.2.3 Active

No events are processed in this state.

3.5.1.2.4 Awaiting Close Response

The following events are processed in this state.

- Receiving an XATMUSER_MTAG_RMCLOSEOK Message
- Receiving Other XATMUSER_MTAG_RMCLOSE Messages
- Connection Disconnected

3.5.1.2.5 Ended

This is the final state.

3.5.1.2.6 State Diagram

The following figure shows the relationship between the CONNTYPE_XATM_OPENONEPIPE initiator states.

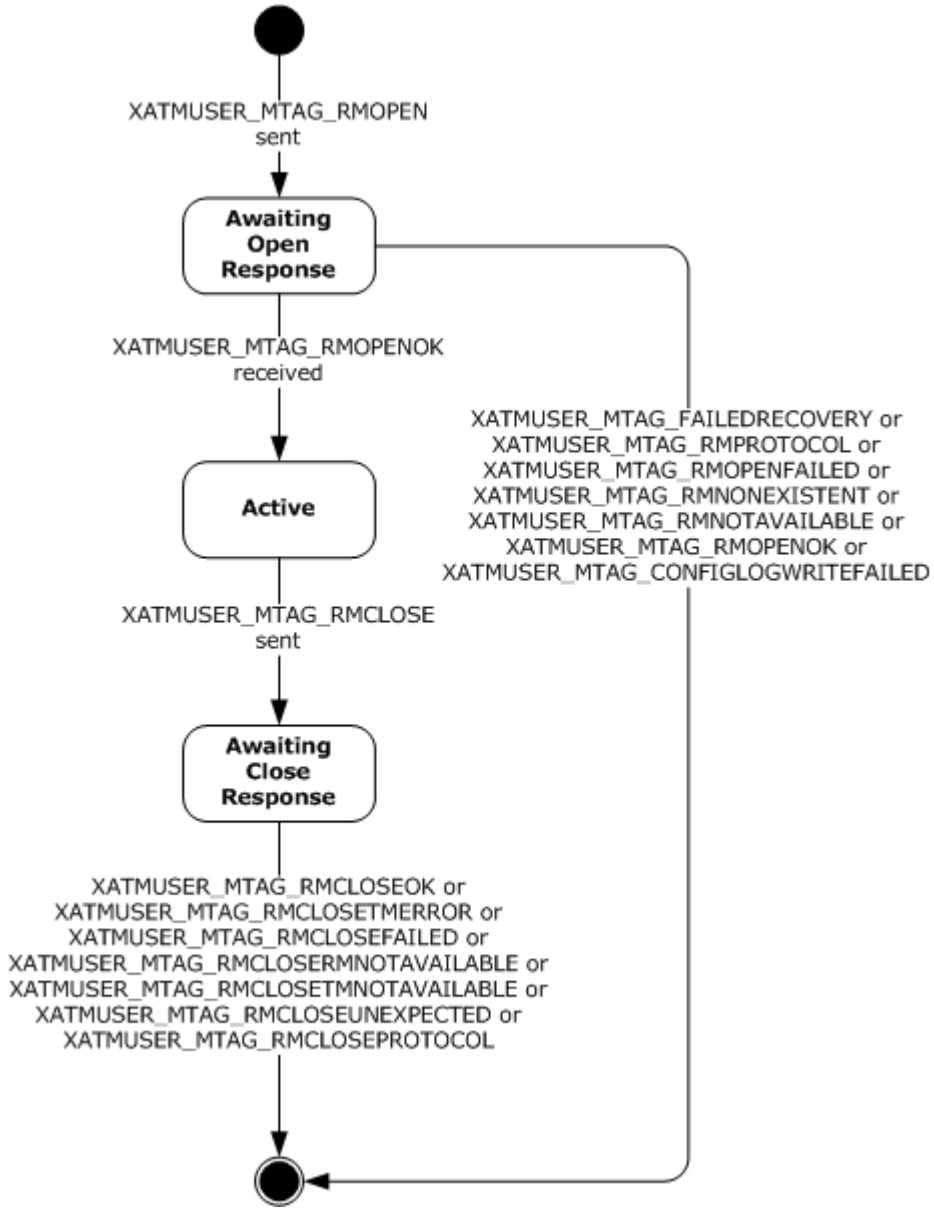


Figure 30: CONNTYPE_XATM_OPENONEPIPE initiator states

3.5.1.3 CONNTYPE_XATM_ENLIST Initiator States

The XA Resource Manager Bridge MUST act as an initiator for the CONNTYPE_XATM_ENLIST connection type. In this role, the XA Resource Manager Bridge MUST provide support for the following states.

- Idle
- Awaiting Enlist Response
- Ended

3.5.1.3.1 Idle

This is the initial state. No events are processed in this state.

3.5.1.3.2 Awaiting Enlist Response

The following events are processed in this state.

-
- Receiving Other XATMUSER_MTAG_ENLIST Messages
-

3.5.1.3.3 Ended

This is the final state.

3.5.1.3.4 State Diagram

The following figure shows the relationship between the CONNTYPE_XATM_ENLIST initiator states.

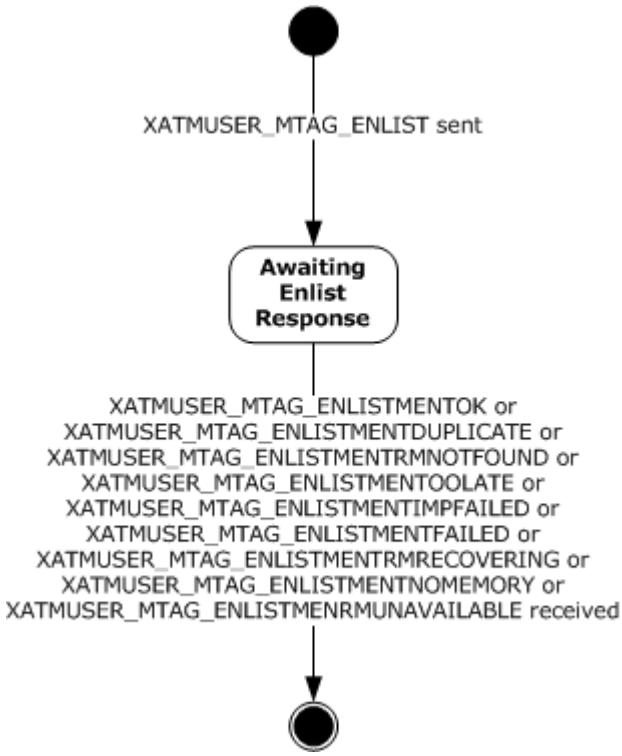


Figure 31: CONNTYPE_XATM_ENLIST initiator states

3.5.2 Timers

None.

3.5.3 Initialization

The XA Resource Manager Bridge SHOULD examine the XA Transactions Enabled flag and perform the following actions.<65>

- If the XA Transactions Enabled flag is not set, the XA Resource Manager Bridge SHOULD refuse to initiate outgoing connections of the following connection types.<66>
  - CONNTYPE_XATM_OPEN
  - CONNTYPE_XATM_OPENONEPIPE
  - CONNTYPE_XATM_ENLIST

3.5.4 Higher-Layer Triggered Events

The operation of the XA Resource Manager Bridge MUST be prepared to process the higher-layer events in this section.

3.5.4.1 Register Two-Pipe XA Resource Manager

This event MUST be signaled by the higher-layer business logic with the following arguments.

- Data Store Name
- DLL Name
- Resource Manager Cookie

If the Register Two-Pipe XA Resource Manager event is signaled, the XA Resource Manager Bridge MUST perform the following actions.

- Attempt to find an XA Resource Manager Proxy object in the Two-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie:
- If an object is found:
  - Return Failure.
- Otherwise:
  - Create a new CONNTYPE_XATM_OPEN connection.
  - Attempt to create a new XA Resource Manager Proxy object with the following settings.
    - The resource manager Global Identifier field is set to NULL GUID
    - The Request Connection field is set to the newly created CONNTYPE_XATM_OPEN connection.
    - The DLL Name field is set to the provided DLL Name.
    - The Data Store Name field is set to the provided Data Store Name.
    - The Pipe field is set to 2.
  - If Successful:

- Set the Connection's XA Resource Manager Proxy reference to the created XA Resource Manager Proxy object.
- Add the newly created XA Resource Manager Proxy to the Two-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie.
- Send an XATMUSER_MTAG_RMOPEN message with the following settings.
  - The **lenDSN** field is set to the length of the provided Data Store Name.
  - The **lenXaDll** field is set to the length of the provided DLL Name.
  - The **Recover** field is set to FALSE.
  - The **DSN** field is set to the provided Data Store Name.
  - The **XaDllFileName** field is set to the provided DLL Name.
- Set the State of Connection referenced by the created XA Resource Manager Proxy object to Awaiting Open Response.

- Otherwise:
  - Return Failure.

3.5.4.2 Unregister Two-Pipe XA Resource Manager

This event MUST be signaled by the higher-layer business logic with the following arguments.

- Resource Manager Cookie

If the Unregister Two-Pipe XA Resource Manager event is signaled, the XA Resource Manager Bridge MUST perform the following actions.

- Attempt to find an XA Resource Manager Proxy object in the Two-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie:
- If an object is found:
  - Remove the XA Resource Manager Proxy object from the Two-Pipe XA Resource Manager Proxy Table maintained by the XA Resource Manager Bridge.
  - If the State of Connection referenced by XA Resource Manager Proxy is set to Active:
    - Disconnect the connection referenced by the found XA Resource Manager Proxy object.
    - Reset the Connection referenced by the found XA Resource Manager Proxy object.
- Otherwise:
  - Return Failure.

3.5.4.3 Enlist Two-Pipe XA Resource Manager

This event MUST be signaled by the higher-layer business logic with the following arguments.

- Resource Manager Cookie
- Transaction object

If the Enlist Two-Pipe XA Resource Manager event is signaled, the XA Resource Manager Bridge MUST perform the following actions.

- Create a new CONNTYPE_XATM_ENLIST connection.
- Attempt to find an XA Resource Manager Proxy object in the Two-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie:
- If an object is found:
  - Construct an XA_XID using the following settings.
    - FormatID set to an implementation specific value.<67>
    - gtridLength set to an implementation specific value.<68>
    - bqualLength set to an implementation specific value.<69>
    - data set to an implementation specific value.<70>
  - Construct an STxInfo, defined in [MS-DTCO] section 2.2.5.10, using the following settings.
    - Set uowTx field to Identifier field of the provided Transaction object
    - SHOULD set tmprotUsed field to 3.<71>
    - Set cbProtocolSpecificTxInfo to 0.
  - Send XATMUSER_MTAG_ENLIST message to Request Connection of XA Resource Manager Proxy with the following arguments.
    - The resource manager Global Identifier field of the found XA Resource Manager Proxy object.
    - The constructed XA_XID.
    - lenImportCookie set to length of constructed STxInfo.
    - ImportCookie set to the constructed STxInfo.
  - Set the State of Connection referenced by XA Resource Manager Proxy object to Awaiting Enlist Response.
- Otherwise:
  - Return Failure.

3.5.4.4 Register One-Pipe XA Resource Manager

This event MUST be signaled by the higher-layer business logic with the following arguments.

- Data Store Name
- DLL Name

- Resource Manager Cookie

If the Register One-Pipe XA Resource Manager event is signaled, the XA Resource Manager Bridge MUST perform the following actions.

- Attempt to find an XA Resource Manager Proxy object in the One-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie:
- If an object is found:
  - Return Failure.
- Otherwise:
  - Create a new CONNTYPE_XATM_OPENONEPIPE connection.
  - Attempt to create a new XA Resource Manager Proxy object with the following values.
    - The resource manager Global Identifier field is set to NULL GUID.
    - The Request Connection field is set to the newly created CONNTYPE_XATM_OPENONEPIPE connection.
    - The DLL Name field is set to the provided DLL name.
    - The Data Store Name field is set to the provided Data Store Name.
    - The Pipe field is set to 1.
  - If successful:
    - Set the XA Resource Manager Proxy referenced by the Connection to the created XA Resource Manager Proxy object.
    - Add the newly created XA Resource Manager Proxy object to the One-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie.
    - Send an XATMUSER_MTAG_RMOPEN message with the following arguments.
      - The **lenDSN** field is set to the length of the provided Data Store Name.
      - The **lenXaDll** field is set to the length of the provided DLL Name.
      - The **Recover** field is set to FALSE.
      - The **DSN** field is set to the provided Data Store Name.
      - The **XaDllFileName** field is set to the provided DLL Name.
    - Set the State of the Connection referenced by the created XA Resource Manager Proxy object to Awaiting Open Response.
  - Otherwise:
    - Return Failure.

3.5.4.5 Unregister One-Pipe XA Resource Manager

This event MUST be signaled by the higher-layer business logic with the following arguments.

- Resource Manager Cookie
- RecoveryNotNecessary flag

If the Unregister One-Pipe XA Resource Manager event is signaled, the XA Resource Manager Bridge MUST perform the following actions.

- Attempt to find an XA Resource Manager Proxy object in the One-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie:
- If an object is found:
  - Remove the XA Resource Manager Proxy object from the One-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge.
  - If the State of Connection referenced by XA Resource Manager Proxy object is set to Active:
    - If the provided RecoveryNotNecessary flag is set to TRUE:
      - Send an XATMUSER_MTAG_RMCLOSE message with the following arguments.
        - The **ShutDownAbrupt** field is set to FALSE.
        - The **Reserved** field is set to 0.
      - Set the State of the Connection referenced by XA Resource Manager Proxy object to Awaiting Close Response.
    - Otherwise:
      - Disconnect the connection referenced by XA Resource Manager Proxy object.
      - Reset the Connection referenced by XA Resource Manager Proxy object.
- Otherwise:
  - Return Failure.

3.5.4.6 Enlist One-Pipe XA Resource Manager

This event MUST be signaled by the higher-layer business logic with the following arguments.

- Resource Manager Cookie
- Transaction Object

If the Enlist One-Pipe XA Resource Manager event is signaled, the XA Resource Manager Bridge MUST perform the following actions.

- Attempt to find an XA Resource Manager Proxy object in the Two-Pipe XA Resource Manager Proxy Table referenced by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie:
  - If an object is found:

- Attempt to enlist the resource manager referenced by the XA Resource Manager Proxy object in the provided Transaction object as described in [MS-DTCO] section 3.5.4.3.

- Otherwise:

  - Return Failure.

3.5.4.7 Create XID

This event MUST be signaled by the higher-layer business logic with the following arguments.

- Transaction object
- Resource Manager Cookie

This event returns a XID.

If the Create XID event is signaled, the XA Resource Manager Bridge MUST perform the following actions.

- Attempt to find an XA Resource Manager Proxy object in the Two-Pipe XA Resource Manager Proxy Table maintained by the XA Resource Manager Bridge keyed by the provided Resource Manager Cookie:

  - If an object is found:

    - Attempt to Create an XID object with the following values.

      - formatID field set to an implementation-specific value.<72>
      - gtridLength set to an implementation specific-value.<73>
      - bqualLength set to an implementation specific-value.<74>
      - data set to an implementation-specific value.<75>

    - Return the XID.

  - Otherwise:

    - Return NULL to indicate failure.

3.5.5 Message Processing Events and Sequencing Rules

3.5.5.1 CONNTYPE_XATM_OPEN as Initiator

For all messages received in this Connection Type, the XA Resource Manager Bridge MUST process the message as specified in section 3.1. The XA Resource Manager Bridge MUST also follow the processing rules specified in the following sections.

3.5.5.1.1 Receiving an XATMUSER_MTAG_RMOPENOK Message

When the XA Resource Manager Bridge receives a XATMUSER_MTAG_RMOPENOK message, it MUST perform the following actions.

- If the State of Connection is set to Awaiting Open Response:

- Set the resource manager Global Identifier field of the XA Resource Manager Proxy object referenced by receiving Connection to the **localRMId** field of the message.
- Set the State of receiving Connection to Active.
- Return success to Register Two-Pipe XA Resource Manager event.

- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.1.2 Receiving Other XATMUSER_MTAG_RMOPEN Messages

When the XA Resource Manager Bridge receives one of the following messages:

- XATMUSER_MTAG_E_RMOPENFAILED
- XATMUSER_MTAG_E_RMNOTEXISTENT
- XATMUSER_MTAG_E_RMNOTAVAILABLE
- XATMUSER_MTAG_E_FAILEDRECOVERY
- XATMUSER_MTAG_E_RMPROTOCOL

The XA Resource Manager Bridge MUST perform the following actions.

- If the State of Connection is set to Awaiting Open Response:
  - Remove the XA Resource Manager Proxy object referenced by receiving Connection from the two-pipe XA Resource Manager Proxy Table.
  - Set the State of receiving Connection to Ended.
  - Return failure to Register Two-Pipe XA Resource Manager event.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.1.3 Connection Disconnected

When a CONNTYPE_XATM_OPEN connection is disconnected, the XA Resource Manager Bridge MUST perform the following actions.

- If the state of Connection is set to Awaiting Open Response:
  - Remove the XA Resource Manager Proxy object referenced by the receiving Connection from the Two-Pipe XA Resource Manager Proxy Table.
  - Set the State of the receiving Connection to Ended.
  - Return failure to Register Two-Pipe XA Resource Manager event.
- Otherwise if the State of the Connection is set to Active:
  - Remove the XA Resource Manager Proxy object referenced by the receiving Connection from the Two-Pipe XA Resource Manager Proxy Table.
  - Set the State of receiving Connection to Ended.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.2 CONNTYPE_XATM_OPENONEPIPE as Initiator

For all messages received in this Connection Type, the XA Resource Manager Bridge MUST process the message as specified in section 3.1. The XA Resource Manager Bridge MUST also follow the processing rules specified in the following sections.

3.5.5.2.1 Receiving an XATMUSER_MTAG_RMOPENOK Message

When the XA Resource Manager Bridge receives a XATMUSER_MTAG_RMOPENOK message, it MUST perform the following actions.

- If the State of the Connection is set to Awaiting Open Response:
  - Set the resource manager Global Identifier field of XA Resource Manager Proxy object referenced by the receiving Connection to **localRMId** field of the message.
  - Set the State field of the receiving Connection to Active.
  - Register with the transaction manager as a Resource Manager as described in section 3.5.4.10 using the **localRMId** field of the message as the Session Identifier.
  - If successful:
    - Set the resource manager referenced by the XA Resource Manager Proxy object to the registered resource manager.
    - Return success to Register One-Pipe XA Resource Manager event.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.2.2 Receiving Other XATMUSER_MTAG_RMOPEN Messages

When the XA Resource Manager Bridge receives one of the following messages:

- XATMUSER_MTAG_E_RMOPENFAILED
- XATMUSER_MTAG_E_RMNOTEXISTENT
- XATMUSER_MTAG_E_RMNOTAVAILABLE
- XATMUSER_MTAG_E_FAILEDRECOVERY
- XATMUSER_MTAG_E_RMPROTOCOL

the XA Resource Manager Bridge MUST perform the following actions.

- If the State of the Connection is set to Awaiting Open Response:
  - Remove the XA Resource Manager Proxy object referenced by the receiving Connection from the One-Pipe XA Resource Manager Proxy Table.
  - Set the State of receiving Connection to Ended.
  - Return failure to Register One-Pipe XA Resource Manager event.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.2.3 Receiving an XATMUSER_MTAG_RMCLOSEOK Message

When the XA Resource Manager Bridge receives a XATMUSER_MTAG_RMCLOSEOK message, it MUST perform the following actions.

- If the State of the Connection is set to Awaiting Close Response:
  - Remove the XA Resource Manager Proxy object referenced by the receiving Connection from the One-Pipe XA Resource Manager Proxy Table.
  - Set the State of the receiving Connection to Ended.
  - Return success to Unregister One-Pipe XA Resource Manager event.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.2.4 Receiving Other XATMUSER_MTAG_RMCLOSE Messages

When the XA Resource Manager Bridge receives one of the following messages:

- XATMUSER_MTAG_E_RMCLOSEFAILED
- XATMUSER_MTAG_E_RMCLOSERMNOTAVAILABLE
- XATMUSER_MTAG_E_RMCLOSETMNOTAVAILABLE
- XATMUSER_MTAG_E_RMCLOSETMERROR
- XATMUSER_MTAG_E_RMCLOSEUNEXPECTED
- XATMUSER_MTAG_E_RMCLOSEPROTOCOL

The XA Resource Manager Bridge MUST perform the following actions.

- If the State of Connection is set to Awaiting Close Response:
  - Remove the XA Resource Manager Proxy object referenced by the receiving Connection from the One-Pipe XA Resource Manager Proxy Table.
  - Set the State of the receiving Connection to Ended.
  - Return success to Unregister One-Pipe XA Resource Manager event.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.2.5 Connection Disconnected

When a CONNTYPE_XATM_OPENONEPIPE connection is disconnected, the XA Resource Manager Bridge MUST perform the following actions.

- If the State of the receiving Connection is set to Awaiting Open Response:
- Remove the XA Resource Manager Proxy object referenced by the receiving Connection from the One-Pipe XA Resource Manager Proxy Table.
- Set the State of the receiving Connection to Ended.
- Return failure to Register One-Pipe XA Resource Manager event.

- Otherwise, if the State of the receiving Connection is set to Awaiting Close Response:
  - Remove the XA Resource Manager Proxy object referenced by the receiving Connection from the One-Pipe XA Resource Manager Proxy Table.
  - Set the State of the receiving Connection to Ended.
  - Return success to Unregister One-Pipe XA Resource Manager event.
- Otherwise if the State of the Connection is set to Active:
  - Remove the XA Resource Manager Proxy object referenced by the receiving Connection from the One-Pipe XA Resource Manager Proxy Table.
  - Set the State of the receiving Connection to Ended.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.3 CONNTYPE_XATM_ENLIST as Initiator

For all messages received in this Connection Type, the XA Resource Manager Bridge MUST process the message as specified in section 3.1. The XA Resource Manager Bridge MUST also follow the processing rules specified in the following sections.

3.5.5.3.1 Receiving an XATMUSER_MTAG_ENLISTMENTOK or an XATMUSER_MTAG_E_ENLISTMENTDUPLICATE Message

When the XA Resource Manager Bridge receives a XATMUSER_MTAG_RMOPENOK or a XATMUSER_MTAG_E_ENLISTMENTDUPLICATE message, it MUST perform the following actions.

- If the State of the Connection is set to Awaiting Enlist Response:
  - Set the State of receiving Connection to Ended.
  - Return success to Enlist Two-Pipe XA Resource Manager event.
- Otherwise the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.3.2 Receiving Other XATMUSER_MTAG_RMENLIST Messages

When the XA Resource Manager Bridge receives one of the following messages:

- XATMUSER_MTAG_E_ENLISTMENTIMPFAILED
- XATMUSER_MTAG_E_ENLISTMENTNOMEMORY
- XATMUSER_MTAG_E_ENLISTMENTFAILED
- XATMUSER_MTAG_E_ENLISTMENTTOOLATE
- XATMUSER_MTAG_E_ENLISTMENTRMRECOVERING
- XATMUSER_MTAG_E_ENLISTMENTRMUNAVAILABLE
- XATMUSER_MTAG_E_ENLISTMENTRMNOTFOUND

the XA Resource Manager Bridge MUST perform the following actions.

- If the State of the Connection is set to Awaiting Enlist Response:
  - Set the State of the receiving Connection to Ended.
  - Return failure to Enlist Two-Pipe XA Resource Manager event.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.5.3.3 Connection Down

When a CONNTYPE_XATM_ ENLIST connection is disconnected, the XA Resource Manager Bridge MUST perform the following actions.

- If the State of the Connection is set to Awaiting Enlist Response:
  - Set the State of receiving Connection to Ended.
  - Return failure to Enlist Two-Pipe XA Resource Manager event.
- Otherwise, the message MUST be processed as an invalid message as specified in section 3.1.5.

3.5.6 Timer Events

None.

3.5.7 Other Local Events

None.

4 Protocol Examples

The following sections describe several examples of common scenarios to illustrate the function of the MSDTC Connection Manager: OleTx XA Transaction Protocol Extension. These protocol examples assume that an OleTx transports session, as specified in [MS-CMPO], has already been established between the two participants.

Participants communicate with each other using OleTx multiplexing connections (as specified in [MS-CMP]) that are in turn layered on top of the OleTx transports infrastructure (as specified in [MS-CMPO]). In these examples, messages are sent from one participant to another by submitting a MESSAGE_PACKET (section 2.2.1.1) to the underlying OleTx multiplexing layer, as specified in [MS-CMP].

4.1 XA Superior Scenarios

These scenarios show how an XA Superior Transaction Manager interoperates with an XA Subordinate Transaction Manager Facet.

4.1.1 Opening an XA Superior Connection with an XA Subordinate Transaction Manager Facet Scenario

This packet sequence is initiated by starting a connection on a transport session between an XA Superior Transaction Manager and an XA Subordinate Transaction Manager Facet.

CONNTYPE_XAUSER_CONTROL: The packet sequence starts when an XA Superior Transaction Manager initiates a connection with an XA Subordinate Transaction Manager Facet using CONNTYPE_XAUSER_CONTROL.

Field	Value	Value description
MsgTag	0x00000005	MTAG_CONNECTION_REQ
fIsMaster	0x00000001	1
dwConnectionId	0x00000001	1
dwUserMsgType	0x00000040	CONNTYPE_XAUSER_CONTROL
dwcbVarLenData	0x00000000	0
dwReserved1	0x00000000	0

The XA Superior Transaction Manager then sends a XA_USER_CONTROL_MTAG_CREATE user message specifying its Resource Manager Recovery GUID (guidXaRM: a9b05f39-2368-4c99-94bc-7b5a4bb3f07d).

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000001	1
dwConnectionId	0x00000001	1
dwUserMsgType	0x00004001	XA_USER_CONTROL_MTAG_CREATE

Field	Value	Value description
dwcbVarLenData	0x00000010	16
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidXaRm	0xA9B05F39	a9b05f39-2368-4c99-94bc-7b5a4bb3f07d
	0x4C992368	
	0x5A7BBC94	
	0x7DF0B34B	

When the XA Subordinate Transaction Manager Facet receives the XA_USER_CONTROL_MTAG_CREATE message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet attempts to add the Resource Manager Recovery GUID to its list of known Resource Manager Recovery GUIDs. If the Resource Manager Recovery GUID is successfully added, then the XA Subordinate Transaction Manager Facet sends a XA_USER_CONTROL_MTAG_CREATED user message to the XA Superior Transaction Manager.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0
dwConnectionId	0x00000001	1
dwUserMsgType	0x00004002	XA_USER_CONTROL_MTAG_CREATED
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

When the XA Superior Transaction Manager gets the XA_USER_CONTROL_MTAG_CREATED response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager maintains this connection to perform transaction recovery and new transactional work with the XA Subordinate Transaction Manager Facet. When the XA Superior Transaction Manager is finished with all transactional work with the XA Subordinate Transaction Manager Facet, it closes the connection by initiating the Disconnect sequence.

4.1.2 Starting an XA Superior Transaction with an XA Subordinate Transaction Manager Facet Scenario

This packet sequence is initiated by starting a connection on a transport session between an XA Superior Transaction Manager and an XA Subordinate Transaction Manager Facet. This scenario assumes that the XA Superior Transaction Manager is maintaining an XA Superior connection with the XA Subordinate Transaction Manager Facet (see section 4.1.1).

CONNTYPE_XAUSER_XACT_START: The packet sequence starts when an XA Superior Transaction Manager initiates a connection with an XA Subordinate Transaction Manager Facet using CONNTYPE_XAUSER_XACT_START.

Field	Value	Value description
MsgTag	0x00000005	MTAG_CONNECTION_REQ
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00000041	CONNTYPE_XAUSER_XACT_START
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

The XA Superior Transaction Manager then sends a XA_USER_XACT_MTAG_START user message. In this example, the XA Superior Transaction Manager specifies its Resource Manager Recovery GUID (guidXaRM: a9b05f39-2368-4c99-94bc-7b5a4bb3f07d), the XID (gtrid:"4f1f5346-e4d2-4ae8-9633-5ab7b8440ef8"; bqual:"0"), and requests an associated transaction with ISOLATIONLEVEL_SERIALIZABLE, no time out, a description of "sample transaction", and ISOFLAG_RETAIN_DONTCARE.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004010	XA_USER_XACT_MTAG_START
dwcbVarLenData	0x000000D4	212
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidXaRm	0xA9B05F39	a9b05f39-2368-4c99-94bc-7b5a4bb3f07d
	0x4C992368	
	0x5A7BBC94	
	0x7DF0B34B	
lenXAIdentifier	0x0000008C	140 (+ 3 bytes of padding)
formatId	0x00000000	0
gtridLength	0x00000024	36
bqualLength	0x00000001	1
Data	0x66316634	gtrid:"4f1f5346-e4d2-4ae8-9633-5ab7b8440ef8"

Field	Value	Value description
	0x36343335	
	0x6434652D	
	0x61342D32	
	0x392D3865	
	0x2D333336	
	0x37626135	
	0x34343862	
	0x38666530	
	0x00000030	bqual:"0"
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

Field	Value	Value description
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
isoLevel	0x00100000	ISOLATIONLEVEL_SERIALIZABLE
dwTimeout	0x00000000	0
szDesc	0x706D6173	sample transaction
	0x7420656C	
	0x736E6172	

Field	Value	Value description
	0x69746361	
	0x00006E6F	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
isoFlags	0x00000005	ISOFLAG_RETAIN_DONTCARE

When the XA Subordinate Transaction Manager Facet receives the XA_USER_XACT_MTAG_START message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet attempts to add the XID to its list of known XIDs.

In this example, the XID is successfully added. The XA Subordinate Transaction Manager Facet then creates an associated OleTx transaction and send a XA_USER_XACT_MTAG_STARTED user message to the XA Superior Transaction Manager specifying the new OleTx transaction identifier (guidTx: 4046037e-9722-46c9-9883-99062341cb35).

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004011	XA_USER_XACT_MTAG_STARTED
dwcbVarLenData	0x00000010	16
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidTx	0x4046037E	4046037e-9722-46c9-9883-99062341cb35
	0x46C99722	
	0x06999883	

Field	Value	Value description
	0x35CB4123	

When the XA Superior Transaction Manager receives the XA_USER_XACT_MTAG_STARTED response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager maintains this connection to perform work on the transaction. When the XA Superior Transaction Manager is finished with all transactional work associated with the transaction, it closes the connection by initiating the Disconnect sequence.

4.1.3 XA Superior Two-Phase Commit Scenario

This scenario shows how an XA Superior Transaction Manager performs the Two-Phase Commit Protocol with an XA Subordinate Transaction Manager Facet. This scenario assumes that the XA Superior Transaction Manager is maintaining an XA Superior connection with the XA Subordinate Transaction Manager Facet (see section 4.1.1).

4.1.3.1 Preparing an XA Superior Transaction with an XA Subordinate Transaction Manager Facet

This packet sequence is initiated by starting a connection on a transport session between an XA Superior Transaction Manager and an XA Subordinate Transaction Manager Facet.

CONNTYPE_XAUSER_XACT_OPEN: The packet sequence starts when an XA Superior Transaction Manager initiates a connection with an XA Subordinate Transaction Manager Facet using CONNTYPE_XAUSER_XACT_OPEN.

Field	Value	Value description
MsgTag	0x00000005	MTAG_CONNECTION_REQ
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00000042	CONNTYPE_XAUSER_XACT_OPEN
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

The XA Superior Transaction Manager then sends a XAUSER_XACT_MTAG_OPEN user message. In this example, the XA Superior Transaction Manager specifies its Resource Manager Recovery GUID (guidXaRM: a9b05f39-2368-4c99-94bc-7b5a4bb3f07d) and the XID (gtrid:"4f1f5346-e4d2-4ae8-9633-5ab7b8440ef8"; bqual:"0").

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2

Field	Value	Value description
dwUserMsgType	0x00004012	XAUSER_XACT_MTAG_OPEN
dwcbVarLenData	0x000000A0	160
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidXaRm	0xA9B05F39	a9b05f39-2368-4c99-94bc-7b5a4bb3f07d
	0x4C992368	
	0x5A7BBC94	
	0x7DF0B34B	
lenXAIdentifier	0x0000008C	140 (+ 3 bytes of padding)
formatId	0x00000000	0
gtridLength	0x00000024	36
bqualLength	0x00000001	1
Data	0x66316634	gtrid:"4f1f5346-e4d2-4ae8-9633-5ab7b8440ef8"
	0x36343335	
	0x6434652D	
	0x61342D32	
	0x392D3865	
	0x2D333336	
	0x37626135	
	0x34343862	
	0x38666530	
	0x00000030	bqual:"0"

Field	Value	Value description
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

Field	Value	Value description
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

When the XA Subordinate Transaction Manager Facet receives the XAUSER_XACT_MTAG_OPEN message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet attempts to locate the XID to its list of known XIDs.

In this example, the XID is located and the XA Subordinate Transaction Manager Facet sends a XA_USER_XACT_MTAG_OPENED user message to the XA Superior Transaction Manager specifying the OleTx transaction identifier (guidTx: 4046037e-9722-46c9-9883-99062341cb35).

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004013	XA_USER_XACT_MTAG_OPENED
dwcbVarLenData	0x00000010	16
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidTx	0x4046037E	4046037e-9722-46c9-9883-99062341cb35
	0x46C99722	
	0x06999883	
	0x35CB4123	

When the XA Superior Transaction Manager receives the XA_USER_XACT_MTAG_OPENED response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager sends a XAUSER_XACT_MTAG_PREPARE user message to the XA Subordinate Transaction Manager Facet specifying that this is not a single-phase commit (fSinglePhase: 0).

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004015	XAUSER_XACT_MTAG_PREPARE
dwcbVarLenData	0x00000004	4
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
fSinglePhase	0x00000000	FALSE: 0

When the XA Subordinate Transaction Manager Facet receives the XAUSER_XACT_MTAG_PREPARE message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet signals the Core Transaction Manager Facet that transaction processing has begun.

In this example, the XA Subordinate Transaction Manager Facet receives Phase One Complete notification from the Core Transaction Manager Facet. In response, the XA Subordinate Transaction Manager Facet sends a XAUSER_XACT_MTAG_REQUEST_COMPLETED user message to XA Superior Transaction Manager indicating that the XA Subordinate Transaction Manager Facet is prepared to commit the transaction.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004017	XAUSER_XACT_MTAG_REQUEST_COMPLETED
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

When the XA Superior Transaction Manager receives the XAUSER_XACT_MTAG_REQUEST_COMPLETED response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager closes the connection by initiating the Disconnect sequence.

4.1.3.2 Committing an XA Superior Transaction with an XA Subordinate Transaction Manager Facet

This packet sequence is initiated by starting a connection on a transport session between an XA Superior Transaction Manager and an XA Subordinate Transaction Manager Facet.

CONNTYPE_XAUSER_XACT_OPEN: The packet sequence starts when an XA Superior Transaction Manager initiates a connection with an XA Subordinate Transaction Manager Facet using CONNTYPE_XAUSER_XACT_OPEN.

Field	Value	Value description
MsgTag	0x00000005	MTAG_CONNECTION_REQ
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00000042	CONNTYPE_XAUSER_XACT_OPEN
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

The XA Superior Transaction Manager then sends a XAUSER_XACT_MTAG_OPEN user message. In this example, the XA Superior Transaction Manager specifies its Resource Manager Recovery GUID (guidXaRM: a9b05f39-2368-4c99-94bc-7b5a4bb3f07d) and the XID (gtrid:"4f1f5346-e4d2-4ae8-9633-5ab7b8440ef8"; bqual:"0").

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004012	XAUSER_XACT_MTAG_OPEN
dwcbVarLenData	0x000000A0	160
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidXaRm	0xA9B05F39	a9b05f39-2368-4c99-94bc-7b5a4bb3f07d
	0x4C992368	
	0x5A7BBC94	
	0x7DF0B34B	
lenXAIdentifier	0x0000008C	140 (+ 3 bytes of padding)
formatId	0x00000000	0
gtridLength	0x00000024	36
bqualLength	0x00000001	1
Data	0x66316634	gtrid:"4f1f5346-e4d2-4ae8-9633-5ab7b8440ef8"
	0x36343335	

Field	Value	Value description
	0x6434652D	
	0x61342D32	
	0x392D3865	
	0x2D333336	
	0x37626135	
	0x34343862	
	0x38666530	
	0x00000030	bqual:"0"
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

Field	Value	Value description
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

When the XA Subordinate Transaction Manager Facet receives the XAUSER_XACT_MTAG_OPEN message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet attempts to locate the XID to its list of known XIDs.

In this example, the XID is located and the XA Subordinate Transaction Manager Facet sends a XA_USER_XACT_MTAG_OPENED user message to the XA Superior Transaction Manager specifying the OleTx transaction identifier (guidTx: 4046037e-9722-46c9-9883-99062341cb35).

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0

Field	Value	Value description
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004013	XA_USER_XACT_MTAG_OPENED
dwcbVarLenData	0x00000010	16
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidTx	0x4046037E	4046037e-9722-46c9-9883-99062341cb35
	0x46C99722	
	0x06999883	
	0x35CB4123	

When the XA Superior Transaction Manager receives the XA_USER_XACT_MTAG_OPENED response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager sends an XAUSER_XACT_MTAG_COMMIT user message to the XA Subordinate Transaction Manager Facet.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004016	XAUSER_XACT_MTAG_COMMIT
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

When the XA Subordinate Transaction Manager Facet receives the XAUSER_XACT_MTAG_COMMIT message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet signals the Core Transaction Manager Facet that the transaction has committed.

When the XA Subordinate Transaction Manager Facet receives Commit Complete notification from the Core Transaction Manager Facet, the XA Subordinate Transaction Manager Facet sends a XAUSER_XACT_MTAG_REQUEST_COMPLETED user message to XA Superior Transaction Manager indicating that the XA Subordinate Transaction Manager Facet has committed the transaction.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0
dwConnectionId	0x00000002	2

Field	Value	Value description
dwUserMsgType	0x00004017	XAUSER_XACT_MTAG_REQUEST_COMPLETED
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

When the XA Superior Transaction Manager receives the XAUSER_XACT_MTAG_REQUEST_COMPLETED response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager closes the connection by initiating the Disconnect sequence.

4.1.4 XA Superior Recovery Scenario

This scenario shows how an XA Superior Transaction Manager performs XA Recovery with an XA Subordinate Transaction Manager Facet. This scenario assumes that the XA Superior Transaction Manager is maintaining an XA Superior connection with the XA Subordinate Transaction Manager Facet (see section 4.1.1).

4.1.4.1 Obtaining a List of XA Superior Transactions to Recover with an XA Subordinate Transaction Manager Facet

This packet sequence uses the XA Superior connection already established with the XA Subordinate Transaction Manager Facet (see section 4.1.1).

To initiate recovery, the XA Superior Transaction Manager sends a XAUSER_CONTROL_MTAG_RECOVER user message. In this example, the XA Superior Transaction Manager specifies to start a new scan (RequestFlags: XARECOVER_START_SCAN) and to return a maximum of five XIDs (totalUOWsRequested: 5).

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000001	1
dwConnectionId	0x00000001	1
dwUserMsgType	0x00004003	XAUSER_CONTROL_MTAG_RECOVER
dwcbVarLenData	0x00000008	8
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
RequestFlags	0x00000001	XARECOVER_START_SCAN
totalUOWsRequested	0x00000005	5

When the XA Subordinate Transaction Manager Facet receives the XAUSER_CONTROL_MTAG_RECOVER message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet starts a scan through its list of known XIDs for any transactions that are prepared but for which the XA Subordinate Transaction Manager Facet does not know outcome. The scan is halted if the number of transactions needing recovery reaches five (totalUOWsRequested: 5).

In this example, the XA Subordinate Transaction Manager Facet locates a total of one transaction in need of recovery. The XA Subordinate Transaction Manager Facet sends a XAUSER_CONTROL_MTAG_RECOVER_REPLY user message to the XA Superior Transaction Manager specifying that there it has one transaction that needs recovery (ulTotalUOWs: 1) and that there are no more recovery records to scan (ReplyFlags: XARECOVER_END_OF_RECS). The message also contains the list of one XID (gtrid:" 4046037e-9722-46c9-9883-99062341cb35"; bqual:"0") in need of outcome.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0
dwConnectionId	0x00000001	1
dwUserMsgType	0x00004005	XAUSER_CONTROL_MTAG_RECOVER_REPLY
dwcbVarLenData	0x00000098	152
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
ReplyFlags	0x00000002	XARECOVER_END_OF_RECS
ulTotalUOWs	0x00000001	1
lenXAIdentifier	0x0000008C	140 (+ 3 bytes of padding)
formatId	0x00000000	0
gtridLength	0x00000024	36
bqualLength	0x00000001	1
Data	0x36343034	gtrid:"4046037e-9722-46c9-9883-99062341cb35"
	0x65373330	
	0x3237392D	
	0x36342D32	
	0x392D3963	
	0x2D333838	
	0x36303939	
	0x31343332	

Field	Value	Value description
	0x35336263	
	0x00000030	bqual:"0"
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

Field	Value	Value description
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

When the XA Superior Transaction Manager receives the XAUSER_CONTROL_MTAG_RECOVER_REPLY response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager iterates through the list of recovery XIDs and locates the corresponding XID to determine the outcome of the transaction.

In this example, the XA Superior Transaction Manager determines that the transaction has aborted and so it initiates a CONNTYPE_XAUSER_XACT_OPEN connection with the XA Subordinate Transaction Manager Facet.

Field	Value	Value description
MsgTag	0x00000005	MTAG_CONNECTION_REQ
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00000042	CONNTYPE_XAUSER_XACT_OPEN
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

The XA Superior Transaction Manager then sends a XAUSER_XACT_MTAG_OPEN user message. In this example, the XA Superior Transaction Manager specifies its Resource Manager Recovery GUID (guidXaRM: a9b05f39-2368-4c99-94bc-7b5a4bb3f07d) and the XID (gtrid:" 4046037e-9722-46c9-9883-99062341cb35"; bqual:"0") that needs recovery.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE

Field	Value	Value description
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004012	XAUSER_XACT_MTAG_OPEN
dwcbVarLenData	0x000000A0	160
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidXaRm	0xA9B05F39	a9b05f39-2368-4c99-94bc-7b5a4bb3f07d
	0x4C992368	
	0x5A7BBC94	
	0x7DF0B34B	
lenXAIdentifier	0x0000008C	140 (+ 3 bytes of padding)
formatId	0x00000000	0
gtridLength	0x00000024	36
bqualLength	0x00000001	1
formatId	0x00000000	0
gtridLength	0x00000024	36
bqualLength	0x00000001	1
Data	0x36343034	gtrid:"4046037e-9722-46c9-9883-99062341cb35"
	0x65373330	
	0x3237392D	
	0x36342D32	
	0x392D3963	
	0x2D333838	
	0x36303939	

Field	Value	Value description
	0x31343332	
	0x35336263	
	0x00000030	bqual:"0"
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

Field	Value	Value description
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	
	0x00000000	

When the XA Subordinate Transaction Manager Facet receives the XAUSER_XACT_MTAG_OPEN message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet attempts to locate the XID to its list of known XIDs.

In this example, the XID is located and the XA Subordinate Transaction Manager Facet sends a XA_USER_XACT_MTAG_OPENED user message to the XA Superior Transaction Manager specifying the OleTx transaction identifier (guidTx: 8f5204b3-5fb9-466a-a0b8-2daf3fcbd9aa).

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004013	XAUSER_XACT_MTAG_OPENED
dwcbVarLenData	0x00000010	16
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64
guidTx	0x8F5204B3	8f5204b3-5fb9-466a-a0b8-2daf3fcbd9aa
	0x466A5FB9	
	0xAF2DA0B8	
	0xAAD9CB3F	

When the XA Superior Transaction Manager receives the XA_USER_XACT_MTAG_OPENED response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager sends a XAUSER_XACT_MTAG_ABORT user message to the XA Subordinate Transaction Manager Facet.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000001	1
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004014	XAUSER_XACT_MTAG_ABORT
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

When the XA Subordinate Transaction Manager Facet receives the XAUSER_XACT_MTAG_ABORT message from the XA Superior Transaction Manager, the XA Subordinate Transaction Manager Facet signals the Core Transaction Manager Facet that the transaction has aborted.

When the XA Subordinate Transaction Manager Facet receives Rollback Complete notification from the Core Transaction Manager Facet, the XA Subordinate Transaction Manager Facet sends a XAUSER_XACT_MTAG_REQUEST_COMPLETED user message to XA Superior Transaction Manager indicating that the XA Subordinate Transaction Manager Facet has aborted the transaction.

Field	Value	Value description
MsgTag	0x00000FFF	MTAG_USER_MESSAGE
fIsMaster	0x00000000	0
dwConnectionId	0x00000002	2
dwUserMsgType	0x00004017	XAUSER_XACT_MTAG_REQUEST_COMPLETED
dwcbVarLenData	0x00000000	0
dwReserved1	0xCD64CD64	dwReserved1: 0xcd64cd64

When the XA Superior Transaction Manager receives the XAUSER_XACT_MTAG_REQUEST_COMPLETED response from the XA Subordinate Transaction Manager Facet, the XA Superior Transaction Manager closes the connection by initiating the Disconnect sequence.

5 Security

5.1 Security Considerations for Implementers

The transaction processing protocol that is defined by this specification is intended for use in an environment where all participants are trusted to collaborate in driving transactions toward a final outcome.

Misuse of this transaction processing protocol can enable participants to perform simple denial of service attacks on their transaction managers. Because transaction managers generally communicate with multiple participants simultaneously, this condition represents a denial of service to other participants.

Consequently, implementers SHOULD take the following steps to ensure that transaction processing occurs in a secure environment:

- Each participant SHOULD<76> initialize MSDTC Connection Manager: OleTx Transports Protocol sessions by using mutual authentication, as specified in [MS-CMPO].
- No transaction remains In Doubt for a longer period of time than the application's higher-layer business logic accepts.

An implementation has the option to further restrict its exposure to security vulnerabilities by initializing the following flag defined in the section 3.1.1 to FALSE:

- XA Transactions Enabled flag

5.2 Index of Security Parameters

None.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Vista
- Windows XP
- Windows Server 2003
- Windows 2000
- Windows NT

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies Windows does not follow the prescription.

<1> Section 2.2.1.1: All versions of Windows set this value to 0xCD64CD64.

<2> Section 2.2.4.1: CONNTYPE_XAUSER_XACT_MIGRATE connection type is not supported by Windows NT 4.0 Option Pack or Windows XP.

<3> Section 2.2.4.1: CONNTYPE_XAUSER_XACT_MIGRATE2 connection type is not supported by Windows NT 4.0 Option Pack, Windows 2000, Windows XP, Windows Server 2003, or Windows Vista. CONNTYPE_XAUSER_XACT_MIGRATE connection type is used instead.

<4> Section 2.2.4.1: CONNTYPE_XAUSER_XACT_BRANCH_START and CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection types are not supported by Windows NT 4.0 Option Pack, Windows 2000, Windows XP, Windows Server 2003, or Windows Vista

<5> Section 2.2.4.1: CONNTYPE_XAUSER_XACT_BRANCH_START and CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection types are not supported by Windows NT 4.0 Option Pack, Windows 2000, Windows XP, Windows Server 2003, or Windows Vista

<6> Section 2.2.4.2.4: When processing an XAUSER_CONTROL_MTAG_RECOVER message, the value of Total UOWs Requested is not verified to be valid on Windows NT 4.0 Option Pack, Windows 2000, or Windows XP. On Windows Server 2003 it is only verified not to be 0.

<7> Section 2.2.4.2.4: When processing an XAUSER_CONTROL_MTAG_RECOVER message, the value used for Total UOWs Requested is set to the minimum value between the provided value and 10000 on Windows Server 2003 RTM.

<8> Section 2.2.4.2.4: The provided Count will be not be minimized by an implementation specific value when processing an xa_recover event on Windows NT 4.0 Option Pack, Windows 2000, or Windows XP. When sending an XAUSER_CONTROL_MTAG_RECOVER message, Windows Implementations will limit the value of totalUOWsRequested and the Count field of the XA Superior Proxy object to a maximum of 5.

<9> Section 2.2.4.2.6: When processing an XAUSER_CONTROL_MTAG_RECOVER_REPLY on Windows NT 4.0 Option Pack, Windows 2000, Windows XP and Windows Server 2003 RTM, the provided Total UoWs Count field is not validated to be less than the number of requested XIDs.

<10> Section 2.2.4.3.1: Setting the Timeout of a transaction created by an xa_start call is not supported on Windows NT 4.0 Option Pack, Windows 2000, Windows XP, and Windows Server 2003. The default value of INFINITE is used.

<11> Section 3.1.3: Setting the XA Transactions Enabled flag to FALSE is not supported by Windows NT 4.0 Option Pack or Windows 2000.

<12> Section 3.2.5.1.2: When processing an XAUSER_CONTROL_MTAG_RECOVER message, the value of Total UOWs Requested is not verified to be valid on Windows NT 4.0 Option Pack, Windows 2000, or Windows XP. On Windows Server 2003 it is only verified not to be 0.

<13> Section 3.2.5.1.2: When processing an XAUSER_CONTROL_MTAG_RECOVER message, the value that Windows implementations use as a maximum for totalUOWsRequested is 10000.

<14> Section 3.2.5.1.2: When processing an XAUSER_CONTROL_MTAG_RECOVER message, the value used for Total UOWs Requested is set to the minimum value between the provided value and 10000 on Windows Server 2003 RTM.

<15> Section 3.2.5.1.2: When processing a XAUSER_CONTROL_MTAG_RECOVER message, the value that Windows implementations use as a maximum for totalUOWsRequested is 10000

<16> Section 3.2.5.1.2: When iterating over available XA Superior Enlistments during processing of an XAUSER_CONTROL_MTAG_RECOVER message on Windows NT 4.0 Option Pack, no check is performed and the actions are taken for all XA Superior Enlistment objects. Under the same conditions, only XA Superior Enlistments in the Preparing State are returned.

<17> Section 3.2.5.2.3: A Connection Down event on a CONNTYPE_XAUSER_XACT_START, CONNTYPE_XAUSER_XACT_OPEN, CONNTYPE_XAUSER_XACT_BRANCH_START, or CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection type is not processed on Windows NT 4.0 Option Pack, Windows 2000, Windows XP, or Windows Server 2003.

<18> Section 3.2.5.3.6: A Connection Down event on a CONNTYPE_XAUSER_XACT_START, CONNTYPE_XAUSER_XACT_OPEN, CONNTYPE_XAUSER_XACT_BRANCH_START, or CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection type is not processed on Windows NT 4.0 Option Pack, Windows 2000, Windows XP, or Windows Server 2003.

<19> Section 3.2.5.5.3: A Connection Down event on a CONNTYPE_XAUSER_XACT_START, CONNTYPE_XAUSER_XACT_OPEN, CONNTYPE_XAUSER_XACT_BRANCH_START, or CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection type is not processed on Windows NT 4.0 Option Pack, Windows 2000, Windows XP, or Windows Server 2003.

<20> Section 3.2.5.6.6: A Connection Down event on a CONNTYPE_XAUSER_XACT_START, CONNTYPE_XAUSER_XACT_OPEN, CONNTYPE_XAUSER_XACT_BRANCH_START, or CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection type is not processed on Windows NT 4.0 Option Pack, Windows 2000, Windows XP, or Windows Server 2003.

<21> Section 3.3.4.3: Disabling Require Thread Affinity is not supported on Windows NT 4.0 Option Pack and Windows XP, and thus Require Thread Affinity is always set to TRUE.

<22> Section 3.3.4.5: CONNTYPE_XAUSER_XACT_MIGRATE connection type is not supported by Windows NT 4.0 Option Pack or Windows XP.

<23> Section 3.3.4.5: CONNTYPE_XAUSER_XACT_MIGRATE connection type is not supported by Windows NT 4.0 Option Pack or Windows XP

<24> Section 3.3.4.7: Setting the Timeout of a transaction created by an xa_start call is not supported on Windows NT 4.0 Option Pack, Windows 2000, Windows XP, and Windows Server 2003. The default value of INFINITE is used.

<25> Section 3.3.4.7: CONNTYPE_XAUSER_XACT_BRANCH_START and CONNTYPE_XAUSER_XACT_BRANCH_OPEN connection types are not supported by Windows NT 4.0 Option Pack, Windows 2000, Windows XP, Windows Server 2003, or Windows Vista

<26> Section 3.3.4.8: Disabling Require Thread Affinity is not supported on Windows NT 4.0 Option Pack and Windows XP, and thus Require Thread Affinity is always set to TRUE.

<27> Section 3.3.4.9: Disabling Require Thread Affinity is not supported on Windows NT 4.0 Option Pack and Windows XP, and thus Require Thread Affinity is always set to TRUE.

<28> Section 3.3.4.9: On Windows NT 4.0 Option Pack, Windows 2000 and Windows XP, the recovery flags to be sent across the wire when processing an xa_recover event will be set to XARECOVER_START_SCAN if TMSTARTRSCAN is set in the flags passed in, XARECOVER_END_SCAN if TMENDRSCAN is set in the flags passed in, and XARECOVER_CONTINUE_SCAN if neither is set in the flags passed in. Also, the Recovery Complete flag is not supported by those operating systems.

<29> Section 3.3.4.9: On Windows NT 4.0 Option Pack, Windows 2000 and Windows XP, the recovery flags to be sent across the wire when processing an xa_recover event will be set to XARECOVER_START_SCAN if TMSTARTRSCAN is set in the flags passed in, XARECOVER_END_SCAN if TMENDRSCAN is set in the flags passed in, and XARECOVER_CONTINUE_SCAN if neither is set in the flags passed in. Also, the Recovery Complete flag is not supported by those operating systems.

<30> Section 3.3.4.9: On Windows NT 4.0 Option Pack, Windows 2000 and Windows XP, the recovery flags to be sent across the wire when processing an xa_recover event will be set to XARECOVER_START_SCAN if TMSTARTRSCAN is set in the flags passed in, XARECOVER_END_SCAN if TMENDRSCAN is set in the flags passed in, and XARECOVER_CONTINUE_SCAN if neither is set in the flags passed in. Also, the Recovery Complete flag is not supported by those operating systems.

<31> Section 3.3.4.9: On Windows NT 4.0 Option Pack, Windows 2000 and Windows XP, the recovery flags to be sent across the wire when processing an xa_recover event will be set to XARECOVER_START_SCAN if TMSTARTRSCAN is set in the flags passed in, XARECOVER_END_SCAN if TMENDRSCAN is set in the flags passed in, and XARECOVER_CONTINUE_SCAN if neither is set in the flags passed in. Also, the Recovery Complete flag is not supported by those operating systems.

<32> Section 3.3.4.9: The Flags provided for the xa_recover event are not validated on Windows NT 4.0 Option Pack, Windows 2000, and Windows XP.

<33> Section 3.3.4.9: The provided Count will be not be minimized by an implementation specific value when processing an xa_recover event on Windows NT 4.0 Option Pack, Windows 2000, or Windows XP. <f>

<34> Section 3.3.4.9: The provided Count will be not be minimized by an implementation specific value when processing an xa_recover event on Windows NT 4.0 Option Pack, Windows 2000, or Windows XP. <f>

<35> Section 3.3.4.10: Disabling Require Thread Affinity is not supported on Windows NT 4.0 Option Pack and Windows XP, and thus Require Thread Affinity is always set to TRUE.

<36> Section 3.3.4.11: CONNTYPE_XAUSER_XACT_MIGRATE connection type is not supported by Windows NT 4.0 or Windows XP.

<37> Section 3.3.4.11: Calling xa_start with TMJOIN will not resume a Suspended transaction on Windows NT 4.0 Option Pack and Windows XP.

<38> Section 3.3.4.11: CONNTYPE_XAUSER_XACT_MIGRATE connection type is not supported by Windows NT 4.0 Option Pack and Windows XP.

<39> Section 3.3.4.11: Disabling Require Thread Affinity is not supported on Windows NT 4.0 Option Pack or Windows XP, and thus Require Thread Affinity is always set to TRUE.

<40> Section 3.3.4.11: CONNTYPE_XAUSER_XACT_MIGRATE connection type is not supported by Windows NT 4.0 Option Pack or Windows XP.

<41> Section 3.3.4.11: Disabling Require Thread Affinity is not supported on Windows NT 4.0 Option Pack or Windows XP, and thus Require Thread Affinity is always set to TRUE.

<42> Section 3.3.5.1.4: Losing connection when waiting for an Abort or Commit response will result in a result of XA_RBCOMMFAIL being returned on Windows NT 4.0 Option Pack and Windows 2000.

<43> Section 3.3.5.1.4: : When sending an XAUSER_CONTROL_MTAG_RECOVER message, Windows Implementations will limit the value of totalUOWsRequested and the Count field of the XA Superior Proxy object to a maximum of 5.

<44> Section 3.3.5.3.7: Losing connection when waiting for an Abort or Commit response will result in a result of XA_RBCOMMFAIL being returned on Windows NT 4.0 Option Pack and Windows 2000.

<45> Section 3.3.5.3.7: : Losing connection when waiting for an Abort or Commit response will result in a result of XA_RBCOMMFAIL being returned on Windows NT 4.0 Option Pack and Windows 2000.

<46> Section 3.3.5.6.8: Losing connection when waiting for an Abort or Commit response will result in a result of XA_RBCOMMFAIL being returned on Windows NT 4.0 Option Pack and Windows 2000.

<47> Section 3.3.5.6.8: Losing connection when waiting for an Abort or Commit response will result in a result of XA_RBCOMMFAIL being returned on Windows NT 4.0 Option Pack and Windows 2000.

<48> Section 3.4.3: Setting the XA Transactions Enabled flag to FALSE is not supported by Windows NT 4.0 Option Pack or Windows 2000.

<49> Section 3.4.3: Setting the XA Transactions Enabled flag to FALSE is not supported by Windows NT 4.0 Option Pack or Windows 2000.

<50> Section 3.4.3.1: On all versions of Windows other than Windows NT 4.0 Option Pack, the XaTmMinWarmRecoveryInterval is initialized to 15 seconds and is configurable to a positive number of seconds.

<51> Section 3.4.3.1: On all versions of Windows other than Windows NT 4.0 Option Pack, the XaTmMinWarmRecoveryInterval is initialized to 15 seconds and is configurable to a positive number of seconds.

<52> Section 3.4.4.1: The Recovery Interval of all XA resource managers is set to 60 seconds on Windows NT 4.0 Option Pack.

<53> Section 3.4.5.1.1: On all versions of Windows other than Windows NT 4.0 Option Pack, Windows 2000, Windows XP RTM, or Windows XP SP1 the provided values of cbDSN or cbXaRmDll in an XATMUSER_MTAG_RMOPEN message are confirmed to be less than 256.

<54> Section 3.4.5.1.1: On all versions of Windows other than Windows NT 4.0 Option Pack, Windows 2000, Windows XP RTM, or Windows XP SP1 the provided values of cbDSN or cbXaRmDll in an XATMUSER_MTAG_RMOPEN message are confirmed to be less than 256.

<55> Section 3.4.5.1.1: The Recovery Interval of all XA resource managers is set to 60 seconds on Windows NT 4.0 Option Pack.

<56> Section 3.4.5.1.1: When an error condition other than those specified occurs during the processing of an XATMUSER_MTAG_RMOPEN message, a message with an invalid MTAG is sent back on Windows NT 4.0 Option Pack.

<57> Section 3.4.5.1.1: If the value of cbDSN or cbXaRmDll in an XATMUSER_MTAG_RMOPEN message is greater than 256 on Windows Server 2008 an XATMUSER_MTAG_E_RMOPENFAILED response message will be sent.

<58> Section 3.4.5.2.1: On all versions of Windows other than Windows NT 4.0 Option Pack, Windows 2000, Windows XP RTM, or Windows XP SP1 the provided values of cbDSN or cbXaRmDll in an XATMUSER_MTAG_RMOPEN message are confirmed to be less than 256.

<59> Section 3.4.5.2.1: On all versions of Windows other than Windows NT 4.0 Option Pack, Windows 2000, Windows XP RTM, or Windows XP SP1 the provided values of cbDSN or cbXaRmDll in an XATMUSER_MTAG_RMOPEN message are confirmed to be less than 256.

<60> Section 3.4.5.2.1: The Recovery Interval of all XA resource managers is set to 60 seconds on Windows NT 4.0 Option Pack.

<61> Section 3.4.7.6: The Recovery Interval of an XA resource manager is not increased after signaling the Recovery Interval Timer on Windows NT 4.0 Option Pack.

<62> Section 3.4.7.6: The Recovery Interval of an XA resource manager is not increased after signaling the Recovery Interval Timer on Windows NT 4.0 Option Pack.

<63> Section 3.4.7.6: The Recovery Interval of an XA resource manager is not increased after signaling the Recovery Interval Timer on Windows NT 4.0 Option Pack.

<64> Section 3.4.7.6: The Recovery Interval of an XA resource manager is not increased after signaling the Recovery Interval Timer on Windows NT 4.0 Option Pack.

<65> Section 3.5.3: Setting the XA Transactions Enabled flag to FALSE is not supported by Windows NT 4.0 Option Pack or Windows 2000.

<66> Section 3.5.3: Setting the XA Transactions Enabled flag to FALSE is not supported by Windows NT 4.0 Option Pack or Windows 2000.

<67> Section 3.5.4.3: When constructing an XA_XID structure, the value that Windows implementations use for formatID is 0x01445443.

<68> Section 3.5.4.3: When constructing an XA_XID structure, the value that Windows implementations use for gtridLength is 16.

<69> Section 3.5.4.3: When constructing an XA_XID structure, the value that Windows implementations use for bqualLength is 32.

<70> Section 3.5.4.3: When constructing an XA_XID structure, the value that Windows implementations use for data is the Identifier field of the Transaction Object, followed by the XA Transaction Manager GUID, followed by the Resource Manager Global Identifier of the XA resource manager Proxy Object.

<71> Section 3.5.4.3: On Windows NT 4.0 Option Pack, the value of tmprotUsed field of an STxInfo is set to 2.

<72> Section 3.5.4.7: When constructing an XA_XID structure, the value that Windows implementations use for formatID is 0x01445443.

<73> Section 3.5.4.7: When constructing an XA_XID structure, the value that Windows implementations use for gtridLength is 16.

<74> Section 3.5.4.7: When constructing an XA_XID structure, the value that Windows implementations use for bqualLength is 32.

<75> Section 3.5.4.7: When constructing an XA_XID structure, the value that Windows implementations use for data is the Identifier field of the Transaction Object, followed by the XA Transaction Manager GUID, followed by the Resource Manager Global Identifier of the XA resource manager Proxy Object.

<76> Section 5.1: Mutual authentication is used by default on Windows XP SP1 and later, Windows Server 2003, Windows Vista, and Windows Server 2008. No authentication is used on Windows NT 4.0 Option Pack, Windows 2000, and Windows XP SP1.

7 Index

T

V

W

X