

# [MS-CSSO]: Collaboration Services System Overview

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do 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. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

This document provides an overview of the Collaboration Services System Overview Protocol Family. It is intended for use in conjunction with the Microsoft Protocol Technical Documents, publicly

available standard specifications, network programming art, and Microsoft Windows distributed systems concepts. It assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A Protocol Family System Document does not require the use of Microsoft programming tools or programming environments in order to implement the Protocols in the System. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

## Abstract

This document describes two sets of logical procedures (Defined Tasks) used when Collaboration Clients locate each other on a network using the directory services provided by an Internet Locator Service (ILS) Server.

These Tasks involve a set of protocols for locating potential collaborative partners. Collaboration itself is performed through instances of the NetMeeting application.

These Tasks require communication between the NetMeeting client and the ILS Server to enable NetMeeting clients to discover other NetMeeting clients available on the server.

This document describes the intended functionality of the Collaboration Services Tasks and how the protocols in this system interact. It provides examples of some of the common user scenarios. It does not restate the processing rules and other details that are specific for each protocol. These details are described in the protocol specifications for each of the protocols and data structures that make up this system.

## Revision Summary

Date	Revision History	Revision Class	Comments
08/14/2009	0.1	Major	First Release.
09/25/2009	0.2	Minor	Updated the technical content.
11/06/2009	0.2.1	Editorial	Revised and edited the technical content.
12/18/2009	0.2.2	Editorial	Revised and edited the technical content.
01/29/2010	1.0	Major	Updated and revised the technical content.
03/12/2010	1.0.1	Editorial	Revised and edited the technical content.
04/23/2010	1.0.2	Editorial	Revised and edited the technical content.
06/04/2010	1.0.3	Editorial	Revised and edited the technical content.
07/16/2010	1.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	1.1	Minor	Clarified the meaning of the technical content.
10/08/2010	1.1	No change	No changes to the meaning, language, or formatting of the technical content.

<b>Date</b>	<b>Revision History</b>	<b>Revision Class</b>	<b>Comments</b>
11/19/2010	1.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	1.1	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	1.1	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	1.1	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	1.1	No change	No changes to the meaning, language, or formatting of the technical content.
06/17/2011	1.2	Minor	Clarified the meaning of the technical content.

# Contents

<b>1</b>	<b>Introduction .....</b>	<b>7</b>
1.1	Glossary .....	7
1.2	References.....	8
1.2.1	Normative References.....	8
1.2.2	Informative References .....	8
<b>2</b>	<b>Overview .....</b>	<b>9</b>
2.1	Summary .....	9
2.2	List of Tasks.....	9
2.2.1	Registering for a Collaboration Session .....	9
2.2.2	Locating a Collaboration Client .....	9
2.3	Relevant Standards.....	9
<b>3</b>	<b>Background Knowledge and System-Specific Concepts .....</b>	<b>10</b>
3.1	Locating a Collaboration Client .....	10
3.2	Internet Locator Service (ILS) .....	10
3.3	Client Registration with ILS .....	11
3.4	ILS Query .....	11
<b>4</b>	<b>Collaboration Client Registration Task .....</b>	<b>12</b>
4.1	Task Overview.....	12
4.1.1	Task Purpose .....	12
4.1.2	Task Applicability .....	12
4.1.3	Task Use Cases .....	12
4.1.3.1	Stakeholders and Interests Summary.....	12
4.1.3.2	Supporting Actors and Task Interests Summary .....	12
4.1.3.3	Use Case Diagrams .....	13
4.1.3.4	Collaboration Client Registration - Client.....	13
4.2	Task Context.....	14
4.2.1	Task Environment .....	14
4.2.2	Task Relationships.....	15
4.2.2.1	Black-Box Relationship Diagram .....	15
4.2.2.2	Task Dependencies .....	15
4.2.2.3	Task Influences .....	15
4.2.3	Task Assumptions and Preconditions.....	15
4.2.4	Task Versioning and Capability Negotiation.....	15
4.3	Task Architecture.....	16
4.3.1	Task Architectural Constraints.....	16
4.3.2	Task Abstract Data Model.....	16
4.3.3	Task Abstract Parameters.....	16
4.3.3.1	Client Registers with an ILS Server .....	17
4.3.3.2	Client Maintains its Registration with an ILS Server.....	17
4.3.3.3	Client Unregisters with an ILS Server .....	17
4.3.4	Task Abstract Results.....	17
4.3.5	White-Box Relationships.....	17
4.3.6	Task Events.....	18
4.3.6.1	Task Timers .....	18
4.3.6.2	Non-Timer Events .....	18
4.3.7	Task Architecture and Communication .....	18
4.3.8	Task Processing Rules .....	19

4.3.8.1	Registration Task Processing Rules .....	19
4.3.8.2	Unregistration Task Processing Rules .....	22
4.3.8.3	Registration Maintenance Task Processing Rules .....	23
4.3.9	Task Failure Scenarios .....	25
4.3.9.1	Registration .....	25
4.3.9.1.1	Failure Scenario: Failure to Bind.....	25
4.3.9.1.2	Failure Scenario: Failure to Add the Collaboration Client's Initial Entry to the ILS.....	26
4.3.9.1.3	Failure Scenario: Failure to Modify the Collaboration Client's Entry in the ILS .....	26
4.3.9.2	Unregistration .....	26
4.3.9.2.1	Failure Scenario: Failure to Bind.....	26
4.3.9.2.2	Failure Scenario: Failure to Delete Collaboration Client's Entry in the ILS ...	27
4.4	Task Details .....	27
4.4.1	Task Precondition Details .....	27
4.4.2	Task Initialization of External Entities.....	27
4.4.3	Task Event Details.....	27
4.4.3.1	Task Timer Details .....	27
4.4.3.2	Task Non-Timer Event Details .....	28
4.4.4	Task Architectural Details.....	28
4.4.5	Task Processing Rule Details.....	28
4.4.5.1	Initialization.....	28
4.4.5.2	Binding to the ILS Server .....	29
4.4.5.3	Adding a Collaboration Client Entry to the ILS Server (Client Registration) .....	29
4.4.5.4	Maintaining the Collaboration Client's Registration Entry on the ILS Server .....	29
4.4.5.5	Unregistering the Collaboration Client from the ILS Server.....	30
4.5	Task Security .....	31
<b>5</b>	<b>Lookup Collaboration Client Task .....</b>	<b>32</b>
5.1	Task Overview.....	32
5.1.1	Task Purpose .....	32
5.1.2	Task Applicability .....	32
5.1.3	Task Use Cases .....	32
5.1.3.1	Stakeholders and Interests Summary.....	32
5.1.3.2	Supporting Actors and Task Interests Summary .....	32
5.1.3.3	Use Case Diagrams.....	32
5.1.3.4	Collaboration Client Query - Client .....	33
5.2	Task Context.....	34
5.2.1	Task Environment .....	34
5.2.2	Task Relationships.....	34
5.2.2.1	Black-Box Relationship Diagrams - Locate Collaboration Client Task.....	35
5.2.2.2	Task Dependencies .....	35
5.2.2.3	Task Influences .....	35
5.2.3	Task Assumptions and Preconditions.....	35
5.2.4	Task Versioning and Capability Negotiation.....	35
5.3	Task Architecture.....	35
5.3.1	Task Architectural Constraints.....	36
5.3.2	Task Abstract Data Model.....	36
5.3.3	Task Abstract Parameters.....	37
5.3.3.1	Client Retrieves Information from ILS Server .....	37
5.3.4	Task Abstract Results.....	37
5.3.5	White-Box Relationships.....	37
5.3.6	Task Events.....	38

5.3.6.1	Task Timers .....	38
5.3.6.2	Task Non-Timer Events .....	38
5.3.7	Task Architecture and Communication .....	38
5.3.8	Task Processing Rules .....	39
5.3.9	Task Failure Scenarios .....	42
5.3.9.1	Failure Scenario: Failure to Bind .....	42
5.3.9.2	Failure Scenario: Failure to Receive Results .....	42
5.4	Task Details .....	42
5.4.1	Task Precondition Details .....	42
5.4.2	Task Initialization of External Entities.....	43
5.4.3	Task Event Details.....	43
5.4.3.1	Task Timer Details .....	43
5.4.3.2	Task Non-Timer Event Details .....	43
5.4.4	Task Architectural Details.....	43
5.4.5	Task Processing Rule Details.....	44
5.4.5.1	Initialization.....	44
5.4.5.2	Binding to the ILS Server .....	44
5.4.5.3	Retrieve List of Available Collaboration Clients from the ILS Server.....	45
5.4.5.4	Retrieve Address of a Collaboration Client of Interest .....	45
5.5	Task Security .....	45
<b>6</b>	<b>Security .....</b>	<b>46</b>
<b>7</b>	<b>Appendix A: Product Behavior .....</b>	<b>47</b>
<b>8</b>	<b>Change Tracking.....</b>	<b>48</b>
<b>9</b>	<b>Index .....</b>	<b>50</b>

# 1 Introduction

A "Defined Task" is a logical procedure that uses one or more Protocols or Systems to accomplish a specific goal. This Defined Task System Document describes two Tasks that provide the ability to locate potential collaboration partners.

In conjunction with Protocol Technical Documents, which are primarily intended to cover Protocols, this Defined Task System Document presents and covers the rules for information exchange relevant to the Tasks, and the Protocols they use, that are used to interoperate or communicate with a Windows client operating system and selected Internet Locator Service (ILS) Server scenarios (those covered in published TDs) in its various environments.

This document describes the Tasks necessary for Collaboration Clients to locate each other on a network using the directory facilities provided by an ILS Server. These two Collaboration Services Tasks are registering the NetMeeting client and querying the ILS Server. Collaboration Services technologies enable people to work together in desktop or browser-based workspaces.

The Tasks in this document address two (or more) Collaboration Clients discovering each other. This document does NOT describe the details of establishing a collaboration session, nor does it describe the Tasks that may take place once such a collaboration session has been established. Readers interested in that phase of the collaborative process will find [\[MS-MNPR\]](#) helpful. The Task of actual communication between the two clients is not covered in this document.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

### **Lightweight Directory Access Protocol (LDAP)**

The following terms are defined in [\[MS-MNPR\]](#):

### **NetMeeting**

The following terms are defined in [\[MS-TAIL\]](#):

**conference**  
**Internet Locator Service (ILS)**  
**session**

The following terms are specific to this document:

**collaborative user (user):** A person using a **Collaboration Client** to communicate with other users.

**Collaboration Client:** Any client wishing to communicate with other clients to exchange information.

**ILS Server:** An abbreviation for **Internet Locator Service (ILS)** Server.

**registration:** The process of registering a **Collaboration Client's** current IP address, along with some additional information, on an **ILS Server**. Other collaborators can locate you through the **ILS**, but can only start a collaboration **session** if the registered address is correct.

The following protocol abbreviations are used in this document:

**TAIL:** Telephony API Internet Locator Service Data Structure.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described in [\[RFC2119\]](#). Note that in [\[RFC2119\]](#) terms, most of these specifications should be imperative, to ensure interoperability. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

Any specification that does not explicitly use one of these terms is mandatory, exactly as if it used MUST.

## 1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 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](mailto: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.

[MS-TAIL] Microsoft Corporation, "[Telephony API Internet Locator Service Protocol Specification](#)".

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

[RFC2252] Wahl, M., Coulbeck, A., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997, <http://www.ietf.org/rfc/rfc2252.txt>

[RFC2256] Wahl, M., "A Summary of the X.500(96) User Schema for use with LDAPv3", RFC 2256, December 1997, <http://www.ietf.org/rfc/rfc2256.txt>

[RFC2589] Yaacovi, Y., Wahl, M., and Genovese, T., "Lightweight Directory Access Protocol (v3): Extensions for Dynamic Directory Services", RFC 2589, May 1999, <http://www.ietf.org/rfc/rfc2589.txt>

### 1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MS-MNPR] Microsoft Corporation, "[Microsoft NetMeeting Protocol Specification](#)".

[LDAP] Microsoft Corporation, "About Lightweight Directory Access Protocol", <http://msdn.microsoft.com/en-us/library/aa366075.aspx>

If you have any trouble finding [LDAP], please check [here](#).



## 2 Overview

Section [1](#), Introduction, primarily describes this Defined Task System Document per se. This section introduces the Tasks that are being documented.

### 2.1 Summary

Software used for collaboration includes applications such as application sharing, e-mail, whiteboarding, sharing a calendar, instant messaging, text chat, and remote desktop. **NetMeeting** is an example of the Microsoft implementation of collaboration software.

One of the common Tasks used in collaboration applications includes methods that allow clients to locate each other. One of the solutions is to register their location in a central store so that other **Collaboration Clients** can find them. This document describes these Tasks as a set of logical procedures (Defined Tasks) that use the Telephony API Internet Locator Service Data block protocol specified in [\[MS-TAIL\]](#) as the key protocol for communications with the central store.

The Defined Tasks described here do not describe the details for establishing a collaboration **session**, nor do they describe the process that takes place after a collaboration session has been established. The logical procedure of communication between two collaboration clients is described in [\[MS-MNPR\]](#).

### 2.2 List of Tasks

The Defined Tasks described in this document are:

Registering for a Collaboration Session: This Task describes Collaboration Client 1 registering, maintaining its **registration**, and unregistering for a session.

Locating a Collaboration Client: This Task describes Collaboration Client 2 locating other Collaboration Clients that have registered with a given **ILS Server**.

#### 2.2.1 Registering for a Collaboration Session

Collaboration Client 1 registers for a collaboration session: Collaboration Client 1 registers with the ILS Server in order to become visible to other users.

Collaboration Client 1 maintains its registration: Collaboration Client 1 refreshes its registration periodically by searching for itself in the **ILS**.

Collaboration Client 1 unregisters from a collaboration session: Collaboration Client 1 unregisters from the ILS, indicating that it is no longer available for a collaboration session.

#### 2.2.2 Locating a Collaboration Client

Collaboration Client 2 queries the ILS: Collaboration Client 2 queries the ILS to locate other Collaboration Clients. After other clients are located, Collaboration Client 2 can use NetMeeting to initiate a collaboration **session**.

### 2.3 Relevant Standards

The Collaboration Services uses the following standards:

**Telephony API Internet Locator Service Protocol**, as specified in [\[MS-TAIL\]](#).

## 3 Background Knowledge and System-Specific Concepts

This section identifies the theoretical and practical information needed to understand this document and the Tasks in this System, and summarizes:

- Background knowledge that is required to understand this document.
- Concepts that are specific to the Tasks in this System, which include the following:
  - Locating a Collaboration Client
  - Internet Locator Service (ILS)
  - Client Registration
  - ILS Query

### 3.1 Locating a Collaboration Client

Typically, collaboration occurs between some pre-defined set of users. Individual collaborators are unlikely to have all the data required to locate and connect to other collaborative users. This leads to a requirement for a well-known shared service for registering a collaboration client and locating other collaboration clients. Specifically, collaboration clients such as NetMeeting rely on IP addresses for communicating with each other. IP addresses are difficult for users to use and then often change over time. Keeping this information in a central store makes it easier for users because the collaboration clients can register themselves and locate each other.

### 3.2 Internet Locator Service (ILS)

The Internet Locator Service (ILS) is the Microsoft software which provides a well-known service capable of storing and searching all data required to establish a collaboration session between two or more users.

ILS provides its services through a protocol based on **LDAP**. The protocol used by the ILS differs from LDAP in follows:

- Uses a unique structure of distinguished names (DNs). NetMeeting puts the most significant elements in the DN first, rather than last: "C=US, O=Microsoft, CN=xxx@abc.com, OBJECTCLASS=rtPerson". By contrast, the standard formatting is: "CN=xxx@abc.com, O=Microsoft, C=US".
- Does not include the required **objectclass** attribute. Instead, it adds an **OBJECTCLASS** element to the end of the DN, as shown in the previous item.
- Does not insert parents into the LDAP server. This is a clear divergence from the LDAP standard, which requires parents to exist before children can be created--that is, to insert the DN: "CN=xxx@abc.com, O=Microsoft, C=US", these DNs must already exist: "O=Microsoft", "C=US".
- Does not understand attribute aliases and is therefore unable to recognize that **sn** and **surname** refer to the same attribute.
- Requires that attributes in a search request be returned in exactly the same order they were requested.
- Specifies "base" scope in search requests, when it really should use "sub", because it is searching for a list of entries, not just one.

- Uses the percent "%" as a wildcard in search requests, rather than the asterisk "\*" specified by the standard.
- In name attributes **surname**, **givenname**, encodes accented European characters as 8-bit ISO 8859-1, rather than multicharacter UTF-8 sequences as required by LDAP ([RFC2252](#) and [RFC2256](#)).
- Uses a nonstandard means of refreshing dynamic entries.

### 3.3 Client Registration with ILS

The first Task detailed in this document is registering and unregistering a collaboration client so that it can be located by a second collaboration client. The client needs to know the hostname or network address of the ILS Server. The client contacts the ILS Server and provides information as specified in section [3.5.3](#) of [\[MS-TAIL\]](#).

### 3.4 ILS Query

When a collaboration client wishes to locate and establish a collaboration session with other clients, it issues an LDAP query request to the ILS in order to locate connection information about available collaboration clients. When it successfully receives the information, it will then be able to establish a collaboration session. The establishment of this collaboration session is beyond the scope of this document.

## 4 Collaboration Client Registration Task

This section describes the Collaboration Client Registration Task, which includes Registration, Registration Maintenance, and Unregistration. This Task is expected to be used by an individual who would like to engage in collaboration activities.

### 4.1 Task Overview

A collaboration client needs to be found by other collaboration clients which wish to collaborate with it. There are three subtasks that describe this process in this Task:

- Registration Task -- To make itself findable, the collaboration client "registers" itself with an ILS Server, by providing the server with contact information such as its IP address, human-friendly information describing the purpose of the collaboration session, and so on.
- Unregistration Task -- When the client no longer wishes to be found, it can Unregister from the ILS Server.
- Registration Maintenance Task -- When this client has registered, it needs to periodically refresh its data with the ILS, in a process known as Registration Maintenance.

#### 4.1.1 Task Purpose

The purpose of this Task is to allow a collaboration client to publish and maintain location information required for other collaboration clients to locate it.

#### 4.1.2 Task Applicability

This Task is applicable when a user wishes to use a well-known service to publish their availability for a collaboration session.

#### 4.1.3 Task Use Cases

##### 4.1.3.1 Stakeholders and Interests Summary

The stakeholders for the Registration Task are collaborative users, collaboration clients, the ILS Server and the **TAIL** protocol.

Collaborative User 1: The a user of the system with the goal of registering a NetMeeting client with the ILS Server.

Collaboration Client 1: In order to participate in a collaboration session, the user must be running a Collaboration Client. In this use case, the client is NetMeeting.

TAIL: The interaction between the Collaboration Client and the ILS Server is performed as specified in [\[MS-TAIL\]](#).

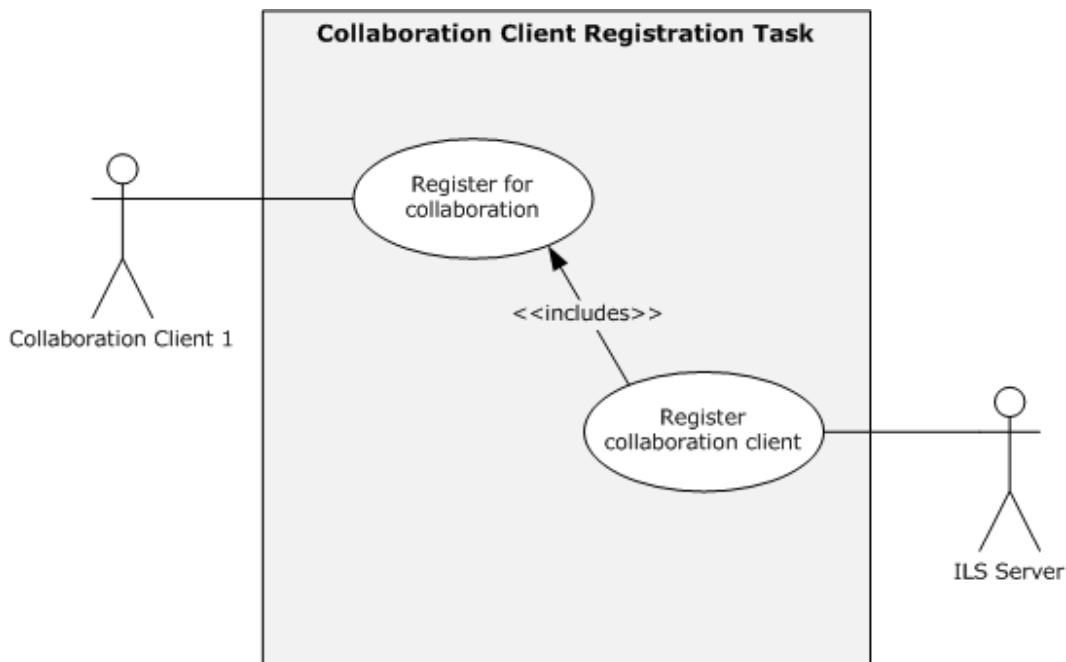
ILS Server: The ILS Server is a computer which runs a name lookup service that provides registration services of Collaboration Clients in order to facilitate the lookup of a list of Collaboration Clients.

##### 4.1.3.2 Supporting Actors and Task Interests Summary

This Task is a standalone Task and is not an actor in any other Task.

### 4.1.3.3 Use Case Diagrams

The following diagram provides an overview of a collaborative user registering for collaboration.



**Figure 1: UML diagram for the Collaboration Client Registration Task**

### 4.1.3.4 Collaboration Client Registration - Client

**Goal:** To allow a Collaboration Client to publish its name and IP address in an ILS Server in order to allow a second Collaboration Client to look up the IP address of the first Collaboration Client. This allows the establishment of a collaboration session between the two Collaboration Clients.

**Context of Use:** User wants to publish the IP address of its Collaboration Client so that the IP address can be found at a later time by another Collaboration Client. All IP addresses must be directly routable.

**Direct Actor:** The direct actor is Collaboration Client 1. The direct actor's interest is to determine the IP address of the computer it is currently executing on and properly register that IP address in an ILS Server.

**Primary Actor:** The primary actor is Collaborative User 1. The primary actor's interest is that the user is indicating they are available to participate in a collaboration session by publishing the IP address of the computer they are executing the direct actor on.

**Supporting Actor:** The supporting actor is the TAIL protocol. TAIL is the protocol that is used to facilitate registration services to an ILS Server.

**Stakeholders and Interests:**

**Collaborative User 1:** A user who wishes to publish their connection information and other collaboration session details to an Internet Locator Service (ILS).

Collaborative User 2: A user who later wishes to initiate a collaboration session hosted by Collaborative User 1.

ILS Server: Stores connection information and other collaboration session details for Collaborative User 1 and makes them available to other users who wish to collaborate with Collaborative User 1.

MS-TAIL: The transport protocol used to get information from Collaborative User 1 to the ILS Server.

Preconditions: A user has determined that they want to be available to participate in a collaboration session with another user.

Minimal Guarantees: The information supplied by Collaboration Client 1 is registered in the ILS Server.

Success Guarantee: The display name and IP address of the computer on which Collaboration Client 1 is executing are registered in the ILS and other Collaboration Clients can query for the location of the collaboration. With the resulting location information, other Collaboration Clients can then optionally initiate a collaboration session with Collaboration Client 1.

Trigger: The primary actor starts the direct actor application which is (optionally) configured to register the host IP address as one of the initial startup activities.

Main Success Scenario:

1. Collaborative User 1 starts Collaboration Client 1 on a workstation.
2. Upon startup, Collaboration Client 1 will attempt to connect to the ILS that it has been configured to use.
3. When a connection to the ILS Server is established, Collaboration Client 1 will then proceed to register the host IP address for the client workstation with the ILS Server.
4. Collaboration Client 1 will then disconnect from the ILS.

Extensions: None.

## **4.2 Task Context**

This section describes the relationship between this Task and its environment.

### **4.2.1 Task Environment**

The typical environment for this Task is a Collaboration Client that has network connectivity to an ILS Server.

To accomplish this Task, the following is required:

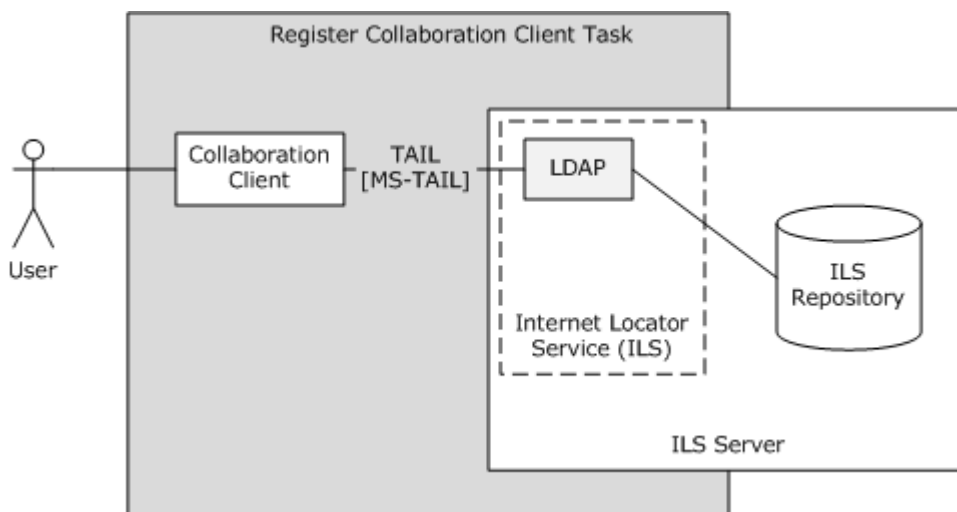
- The Collaboration Client is configured on a Microsoft Windows® client computer running Microsoft Windows® 2000 operating system, Windows® XP operating system, or Windows Server® 2003 operating system.
- The ILS Server is available over an IP network.
- The system requires a networked environment in which collaboration members are connected.
- The networked environment has to be capable of supporting TCP/IP traffic such as TCP and UDP.

- Firewall ports, network translation, and network devices such as routers and switches are required to be configured in order that collaboration network traffic can flow between collaboration members and the ILS Server.

## 4.2.2 Task Relationships

The only entity this Task interacts with is an ILS Server.

### 4.2.2.1 Black-Box Relationship Diagram



**Figure 2: Black-box diagram for the Collaboration Client Registration Task**

### 4.2.2.2 Task Dependencies

Dependencies for the Collaboration Client Registration Task are as follows:

- Collaboration Client application software is NetMeeting version 3.0 which is included in Windows® XP operating system and Windows Server® 2003 operating system.
- ILS Server is an optional component of Microsoft Windows® 2000 Server operating system.

### 4.2.2.3 Task Influences

There are no external influences for this Task.

## 4.2.3 Task Assumptions and Preconditions

This Task has the following assumptions and preconditions:

- The location of the ILS configured manually by the user within the Collaboration Client.
- Installation and network availability over TCP/IP of the ILS Server required.

## 4.2.4 Task Versioning and Capability Negotiation

There is no versioning or capability negotiation for this Task.

### 4.3 Task Architecture

The following diagram illustrates Collaboration Client 1 registering itself with the ILS Server.



**Figure 3: High-Level architecture diagram of the Collaboration client Registration Process**

#### 4.3.1 Task Architectural Constraints

There **MUST** be an ILS Server available that all clients participating in a collaboration session must be aware of. The various collaboration clients will use this ILS Server to register and locate one another throughout the network. Each client **MUST** know the name/address of the ILS Server in advance.

#### 4.3.2 Task Abstract Data Model

This section describes state established, used, and maintained by processing rules of this Task. State may be volatile or persisted. State may pertain to one, some, or all instances of the Task. The Task's state consists of the values of the named data elements (also called state variables) presented in this section. The overall organization of the data elements, with their names, is the Abstract Data Model. It is intended to facilitate the reader's conceptual understanding of the specification. While a Task's processing rules may depend upon associations established by the structure of its Abstract Data Model, such association can be achieved in other ways. Implementations may depart from this model so long as their external behavior remains consistent with that described in this document.

Each collaboration client that wants to be located by other collaboration clients will register itself in the ILS Server. The ILS Server in turn will store this information within its repository system for fast data retrieval. Each new client wanting to initiate a conferencing session will retrieve the **conference** information from the ILS Server. With the location information retrieved, the clients are then able to establish a direct connection with each other.

Each collaboration client registers itself with the ILS as specified in [\[MS-TAIL\]](#).

#### 4.3.3 Task Abstract Parameters

This section describes data passed to an instance of this Task at the time it is invoked or triggered. The parameters consist of the values of the named data elements presented in this section. The organization of a data element, with its names, is an Abstract Parameter. It is intended to facilitate the reader's conceptual understanding of the specification. While a Task's processing rules may depend upon associations established by the structure of its Abstract Parameters, such association can be achieved in other ways. Implementations may depart from this abstraction so long as their external behavior remains consistent with that described in this document.



When a client registers with the ILS Server, it sends information describing itself. This information will be used by other clients to identify and locate the client it desires to collaborate with. This information is transferred in LDAP packets and utilizes the LDAP Extension described in [\[RFC2589\]](#). The following section describes the type of information sent to and retrieved from the ILS Server.

#### **4.3.3.1 Client Registers with an ILS Server**

When a client registers with an ILS Server, it registers itself as an RtPerson using the object class definition detailed in [\[MS-TAIL\] \(section 2.2.3\)](#).

#### **4.3.3.2 Client Maintains its Registration with an ILS Server**

When a client maintains its registration with an ILS Server, it issues an LDAP query to the ILS Server looking for itself.

#### **4.3.3.3 Client Unregisters with an ILS Server**

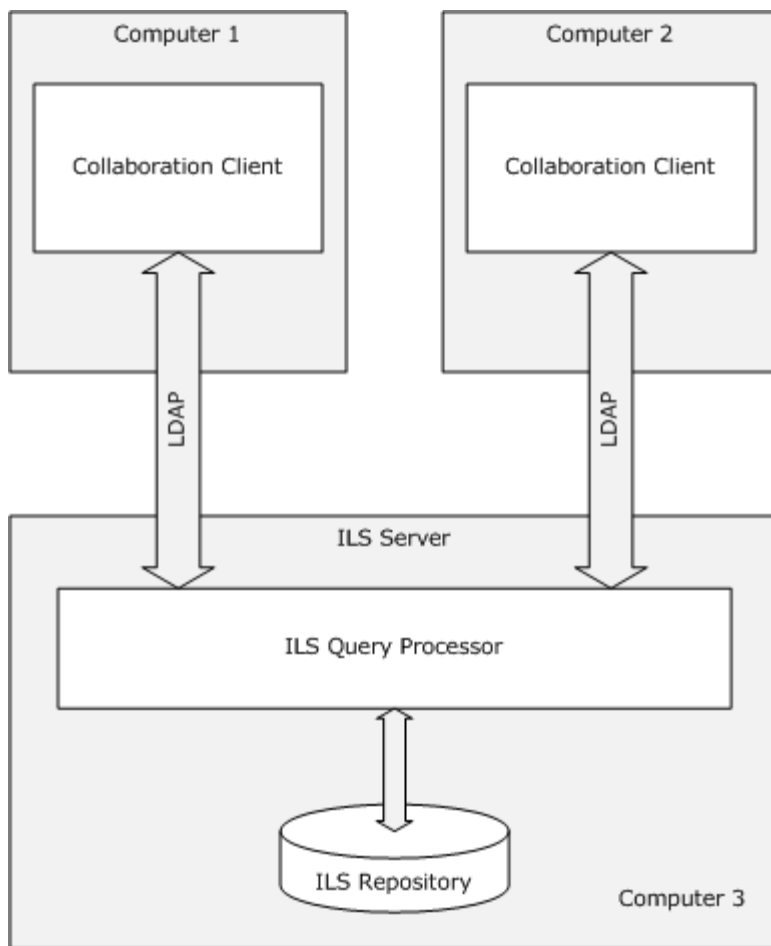
When a client terminates, it unregisters with the ILS Server using the LDAP Delete Operation followed by the LDAP Unbind Operation.

#### **4.3.4 Task Abstract Results**

This section describes data returned by an instance of this Task to its caller. The results consist of the values of the named data elements presented in this section. The organization of a data element, with its names, is an Abstract Result. It is intended to facilitate the reader's conceptual understanding of the specification. While a Task's processing rules may depend upon associations established by the structure of its Abstract Results, such association can be achieved in other ways. Implementations may depart from this abstraction so long as their external behavior remains consistent with that described in this document.

#### **4.3.5 White-Box Relationships**

The following illustration outlines the various components present whenever ILS directory services are requested.



**Figure 4: White-box diagram for the Collaboration Client Registration Task**

#### 4.3.6 Task Events

##### 4.3.6.1 Task Timers

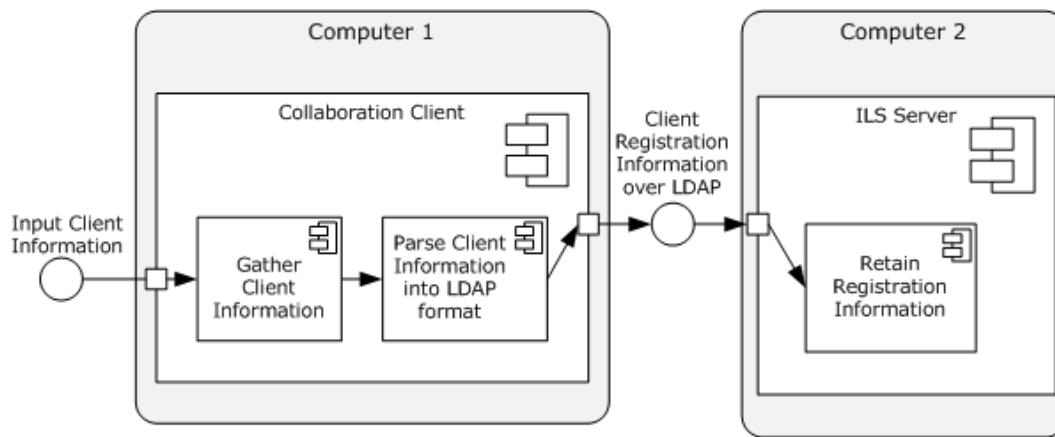
In order for the client's information to be retained on the ILS Server, a refresh must occur within a specific timeframe of the initial Registration Task.

##### 4.3.6.2 Non-Timer Events

In the event of abnormal disconnect between the client and ILS Server, the ILS Server will retain the client's registration information for a short duration of time.

#### 4.3.7 Task Architecture and Communication

The following diagram illustrates the various components required when registering a Collaboration Client.



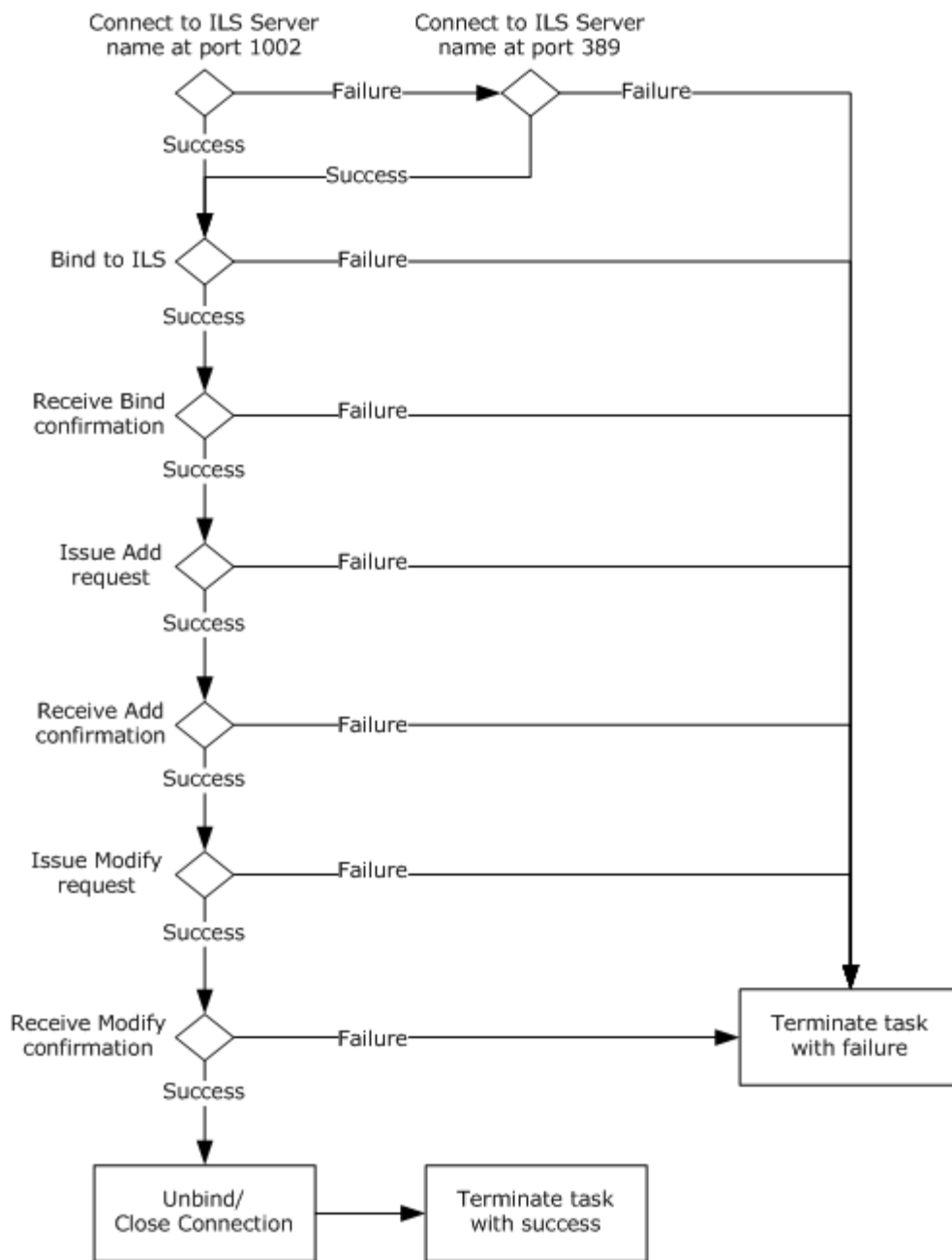
**Figure 5: Communication architecture diagram for the Collaboration Client Registration Task**

Computer 1 will gather the registration information from various sources, including the user. When the information has been gathered, the client transmits the data to the ILS for storage, following the protocols specified in [\[MS-TAIL\]](#). This information will be retained on the ILS Server. This process will allow other collaboration clients throughout the network to locate Computer 1 when a collaboration session is desired.

### 4.3.8 Task Processing Rules

#### 4.3.8.1 Registration Task Processing Rules

The following flowchart depicts the operational flow of the Registration portion of the Collaboration Client Registration Task.



**Figure 6: Collaboration Client Registration Task processing rules flowchart**

The abstract parameter, preconditions, and main success scenario of the Registration Task are described in the following sections.

#### Abstract Parameters

- Name of ILS Server (SERVER)
- Name of entry to create/delete in the ILS (ENTRY NAME)

## Preconditions

None.

## Main Success Scenario

1. Bind to the ILS using LDAP.
  1. Resolve the name of the ILS Server to a network address if necessary. (The server location may have been given as a network address to the Collaboration Client.)
  2. Bind to the address, using port 1002. This establishes the LDAP connection.
  3. Receive Bind confirmation.
  4. Go to 2.
2. Add the Entry to the ILS.
  1. Issue an Add request for the dynamic entry, using the ENTRY NAME.
  2. Receive Add confirmation.
  3. Go to 3.
3. Modify the new entry in the ILS.
  1. Issue a Modify request to complete the data in the new entry.
  2. Receive Modify confirmation.
  3. Go to 4.
4. Unbind from the ILS.
  1. Issue an Unbind request to the ILS.
  2. Go to 5.
5. Close Connection.
6. Terminate Task with success.

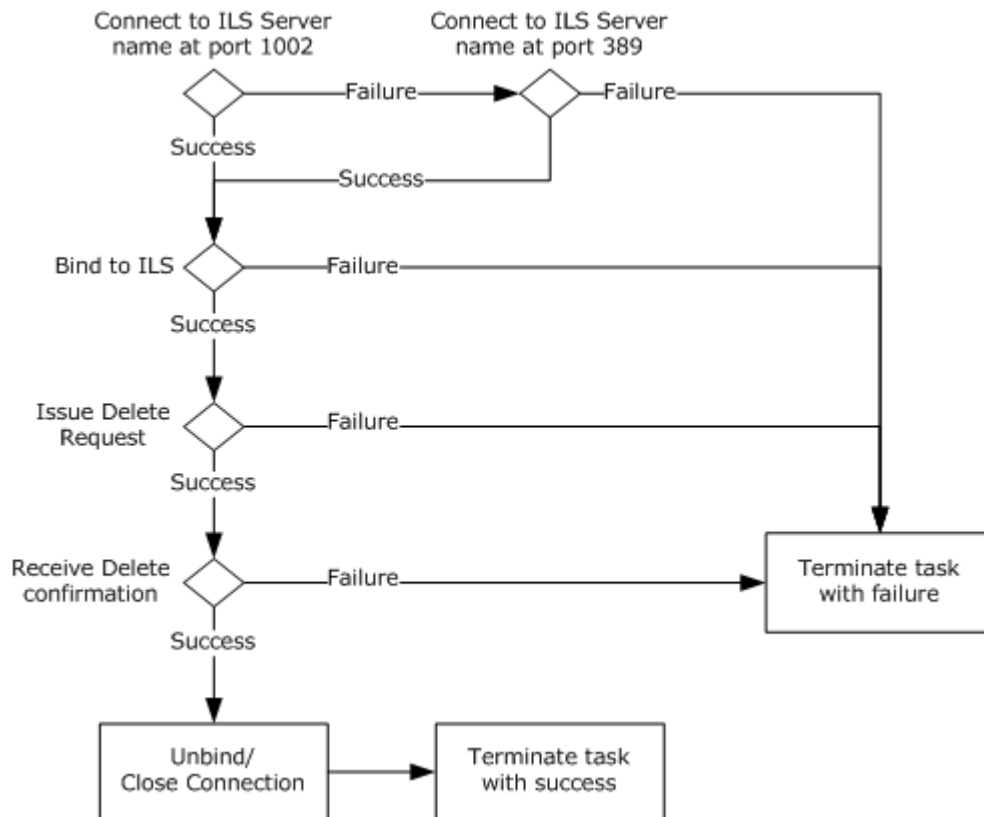
## Extensions:

- 1.a.1 Unable to resolve name.
- 1.a.1.a Terminate Task with error.
- 1.b.1 Unable to bind to address.
- 1.b.1.a Retry using port 389.
- 1.b.1.a.1 Unable to bind to address.
- 1.b.1.a.1.a Terminate Task with error.
- 1.c.1 Did not receive response from server.
- 1.c.1.a Terminate Task with error.

- 2.b.1 Did not receive confirmation from server.
- 2.b.1.a Close connection.
- 2.b.2.b Terminate Task with error.
- 3.b.1 Did not receive confirmation from server.
- 3.b.1.a Go to 2.b.1.a.

#### 4.3.8.2 Unregistration Task Processing Rules

The following diagram shows the operation flow for the Unregister Task:



**Figure 7: Unregistration Task processing rules flowchart**

The abstract parameter, preconditions, and main success scenario of the Unregistration Task are described in the following sections.

##### Abstract Parameters

- Name of ILS Server (SERVER)
- Name of entry to create/delete in the ILS (ENTRY NAME)

##### Preconditions

An entry for the Collaboration Client exists in the ILS.

#### Main Success Scenario

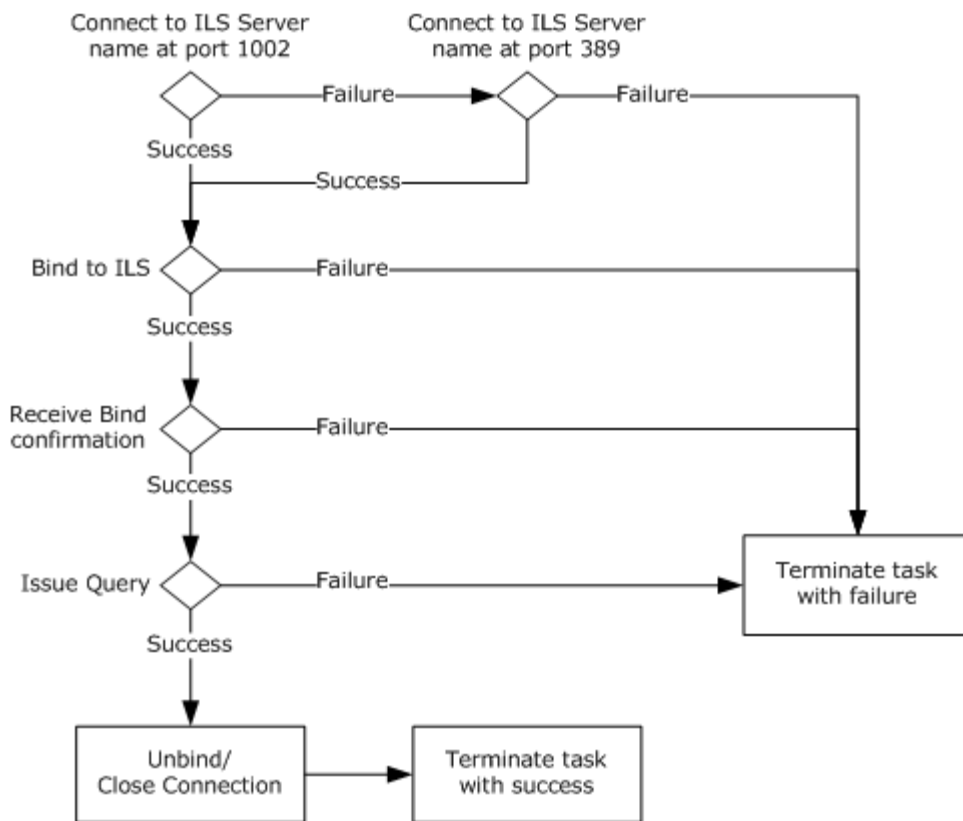
1. Bind to the ILS using LDAP.
  1. a. Resolve the name of the ILS Server to a network address if necessary. (The server location may have been given as a network address to the collaboration client.)
  2. Bind to the address, using port 1002. This establishes the LDAP connection.
  3. Receive bind confirmation.
  4. Go to 2.
2. Delete the entry from the ILS.
  1. Issue a Delete request for the dynamic entry, using the ENTRY NAME.
  2. Receive Delete confirmation.
  3. Go to 3.
3. Unbind from the ILS.
  1. Issue an Unbind request to the ILS.
  2. Go to 4.
4. Close Connection.
5. Terminate Task with success.

#### Extensions:

- 1.a.1 Unable to resolve name.
- 1.a.1.a Terminate Task with error.
- 1.b.1 Unable to bind to address.
- 1.b.1.a Retry using port 389.
- 1.b.1.a.1 Unable to bind to address.
- 1.b.1.a.1.a Terminate Task with error.
- 1.c.1 Did not receive response from server.
- 1.c.1.a Terminate Task with error.
- 2.b.1 Did not receive confirmation from server.
- 2.b.1.a Go to 4.

#### 4.3.8.3 Registration Maintenance Task Processing Rules

The following diagram shows the operation flow for the Registration Maintenance Task.



**Figure 8: Registration Maintenance Task processing rules flowchart**

The abstract parameter, preconditions, and main success scenario of the Unregistration Task are described in the following sections.

#### Abstract Parameters

- Name of ILS Server (SERVER)
- Name of entry to refresh in the ILS (ENTRY NAME)

#### Preconditions

The entry to be refreshed must exist in the ILS.

The abstract parameter, preconditions, and main success scenario of the Registration Task are described in the following sections.

#### Main Success Scenario

1. Bind to the ILS using LDAP.
  1. Resolve the name of the ILS Server to a network address if necessary. (The server location may have been given as a network address to the collaboration client.)
  2. Bind to the address, using port 1002. This establishes the LDAP connection.
  3. Receive Bind confirmation.



4. Go to 2.
2. Issue Query.
  1. Issue a Query request for the dynamic entry using the ENTRY NAME.
  2. Receive Query confirmation.
  3. Go to 3.
3. Unbind from the ILS.
  1. Issue an Unbind request to the ILS.
  2. Go to 4.
4. Close Connection.
5. Terminate Task with success.

Extensions:

- 1.a.1 Unable to resolve name.
- 1.a.1.a Terminate Task with error.
- 1.b.1 Unable to bind to address.
- 1.b.1.a Retry using port 389.
- 1.b.1.a.1 Unable to bind to address.
- 1.b.1.a.1.a Terminate Task with error.
- 1.c.1 Did not receive response from server.
- 1.c.1.a Terminate Task with error.
- 2.b.1 Did not receive confirmation from server.
- 2.b.1.a Close connection.
- 2.b.2.b Terminate Task with error.
- 3.b.1.a Go to 2.b.1.a.

#### **4.3.9 Task Failure Scenarios**

##### **4.3.9.1 Registration**

###### **4.3.9.1.1 Failure Scenario: Failure to Bind**

In this scenario, Collaboration Client 1 fails to connect to the ILS.

Causes: This may be due to network failures, server failure, or a failure of the ILS to respond to the client's request.

Consequences: Failure of the Registration Task. There are no consequences to the ILS.

Recognition: The failure is recognized by the Collaboration Client a) being unable to connect to the ILS at a network level, or b) failing to receive a Bind acknowledgement from the ILS.

Task Recovery: None.

Consequences Of Recovery: None.

#### **4.3.9.1.2 Failure Scenario: Failure to Add the Collaboration Client's Initial Entry to the ILS**

Causes: Potential causes are a) network failure, b) server failure, c) ILS process failure, or d) some other unspecified error.

Consequences: Failure of the Registration Task. There is a possibility that the ILS has indeed created the new entry, but was unable to notify the Collaboration Client. In this latter case, the newly created entry will be deleted upon expiration of its time to live (TTL).

Recognition: The Collaboration Client fails to receive the Add acknowledgement.

Task Recovery: None.

Consequences of Recovery: None.

#### **4.3.9.1.3 Failure Scenario: Failure to Modify the Collaboration Client's Entry in the ILS**

Causes: Potential causes are a) network failure, b) server failure, c) ILS process failure, or d) some other unspecified error.

Consequences: Failure of the Registration Task. In this case, the Collaboration Client's entry exists in the ILS, and may have been modified, but the client is unaware of the modification. The Collaboration Client's entry will be deleted upon expiration of its TTL.

Recognition: The Collaboration Client fails to receive the Modify acknowledgement.

Task Recovery: None.

Consequences of Recovery: None.

### **4.3.9.2 Unregistration**

#### **4.3.9.2.1 Failure Scenario: Failure to Bind**

Causes: Potential causes are a) network failure, b) server failure, c) ILS process failure, or d) some other unspecified error.

Consequences: Failure of the Unregistration Task. If the Collaboration Client's entry still exists in the ILS, it will be deleted upon expiration of its TTL.

Recognition: The failure is recognized by the Collaboration Client a) being unable to connect to the ILS at a network level, or b) failing to receive a Bind acknowledgement from the ILS.

Task Recovery: None.

Consequences of Recovery: None.

#### **4.3.9.2.2 Failure Scenario: Failure to Delete Collaboration Client's Entry in the ILS**

Causes: Some potential causes are a) network failure, b) server failure, c) ILS process failure, or d) standard LDAP error conditions.

Consequences: Failure of the Unregistration Task. If the Collaboration Client's entry still exists in the ILS, it will be deleted upon expiration of its TTL.

Recognition: The failure is recognized by the Collaboration Client a) being unable to connect to the ILS at a network level, or b) failing to receive a Delete acknowledgement from the ILS.

Task Recovery: None.

Consequences of Recovery: None.

### **4.4 Task Details**

This section contains the details that complete the descriptions in earlier sections of the document. These are needed to understand and implement this Task.

#### **4.4.1 Task Precondition Details**

The Collaboration Client does not automatically discover the ILS Server. The location of the ILS Server is entered manually by the user within the Collaboration Client application. This is normally carried out upon application startup where some type of user dialog is presented. At that point, the user will input the name/address of the ILS Server to use during the registration process. This information will be used in establishing a connection to the ILS Server.

#### **4.4.2 Task Initialization of External Entities**

The system requires a networked environment in which collaboration members are connected.

The networked environment has to be capable of supporting TCP/IP traffic such as TCP and UDP.

Firewall ports, network translation and network devices such as routers and switches are required to be configured so that collaboration network traffic can flow between collaboration members.

#### **4.4.3 Task Event Details**

##### **4.4.3.1 Task Timer Details**

In order for the client's location information to be retained on the ILS Server, a refresh must occur at 10 minute intervals starting with the initial Registration. The refresh is performed from the client as an LDAP Search operation. This will notify the ILS Server that the client is still active. A timer is utilized on the client to ensure that this operation occurs within a 10 minute timeframe.

In the event that a client connection is abnormally dropped, the ILS Server will retain the client's registration information for at least a configurable duration of time. See section [3.2](#) in [\[MS-TAIL\]](#). After the duration of time has elapsed, the client's registration information MAY be removed from the ILS repository.

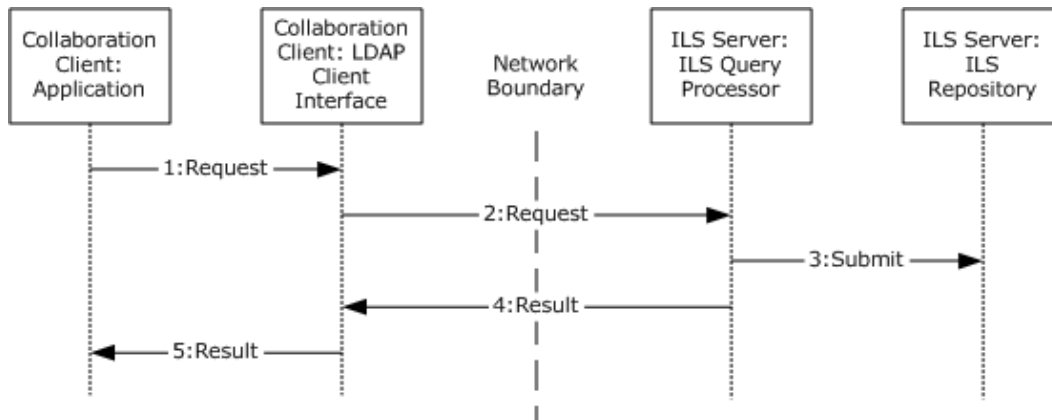
In the event that the LDAP Search Operation above did not find the collaboration's client registration entry, the client will need to re-register its information.

#### 4.4.3.2 Task Non-Timer Event Details

There are no non-timer events in this Task.

#### 4.4.4 Task Architectural Details

This section illustrates an example of a Collaboration Client sending registration information to an ILS Server. The client utilizes several LDAP operations in order to accomplish the request see [\[LDAP\]](#).



**Figure 9: Communication architecture details for the Collaboration Client Task**

The sequence flow for the Collaboration Client Registration Task is as follows:

1. Request: The initial connect process is carried out using LDAP initialization. The name of the target ILS Server is supplied at this point along with the TCP port to use (by default, port 1002). Next the client will use the LDAP Bind Operation associated itself with the ILS Server. When the Bind is successful, the client creates the requisite entry through an LDAP Add operation, and then completes all the additional information in the entry through an LDAP Modify operation. Details of the information used in this Task can be found in [\[MS-TAIL\]](#).
2. Request: LDAP will send this information over the network to the destination ILS Server. The ILS Server will then pick up this information and validate it.
3. Submit: The client registration data will then be placed into the ILS Repository typically as cached data.
4. Result: The success/failure result of the submit operation will be sent back to the client.
5. Result: The client will then review the results provided by the LDAP client functions and take appropriate action.

#### 4.4.5 Task Processing Rule Details

This section describes the processing rules required to register a Collaboration Client with the ILS Server.

##### 4.4.5.1 Initialization

The first process is to initialize the subsystems used by the Collaboration Client application.

The LDAP layer is initialized and the destination ILS Server defined. This is accomplished by using the LDAP Initialization Operation. The HostName (or target ILS Server) along with the TCP port is specified.

#### 4.4.5.2 Binding to the ILS Server

This operation is required to allow the client to "bind" itself to a specific entry (or create a new entry) in the ILS Server. The LDAP Bind Operation is used to accomplish this.

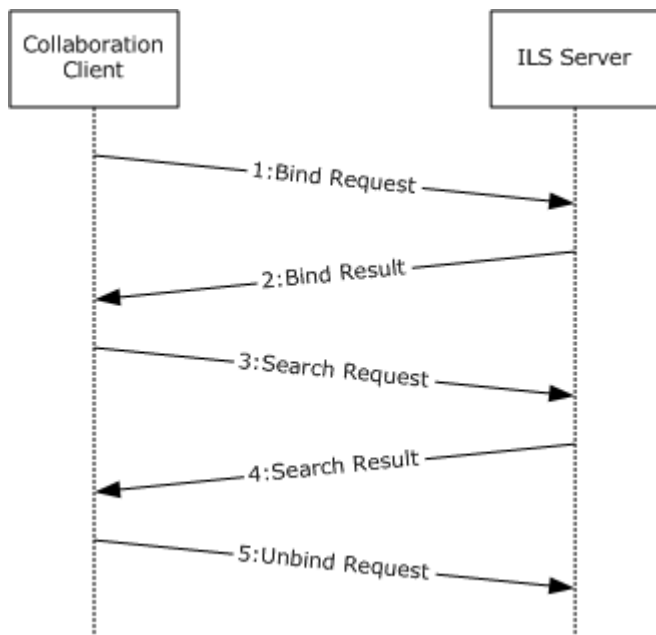
#### 4.4.5.3 Adding a Collaboration Client Entry to the ILS Server (Client Registration)

When the initialization phase is complete, the next step is to add the Collaboration Client's registration information as described in section [4.3.3.1](#). This is accomplished using the LDAP Add Operation followed by the LDAP Modify Operation.

#### 4.4.5.4 Maintaining the Collaboration Client's Registration Entry on the ILS Server

The ILS Server will retain the Collaboration Client's registration information for a short duration of time (typically 10 minutes). If the Collaboration Client does not "refresh" the ILS Server within this time frame, then the ILS Server MAY delete the entry from its repository.

The "refresh" is accomplished by issuing an LDAP Search Operation. The DN will consist of the string "objectClass=rtPerson", the search should be restricted to the base entry, the search filter will consist of the entry to search for (for example "(&(objectClass=rtPerson)(cn=<email\_address>)(sttl=10))") where <email\_address> is the Common Name (CN) to search for, and the attributes requested will consist of one entry of *sttl*. It should be noted that the filter item "(sttl=10)" will attempt to reset the TTL (Time-To-Live) value on the ILS Server to 10 minutes.



**Figure 10: Sequence diagram of Collaboration Client refreshing its Time-To-Live (TTL) entry on an ILS Server**

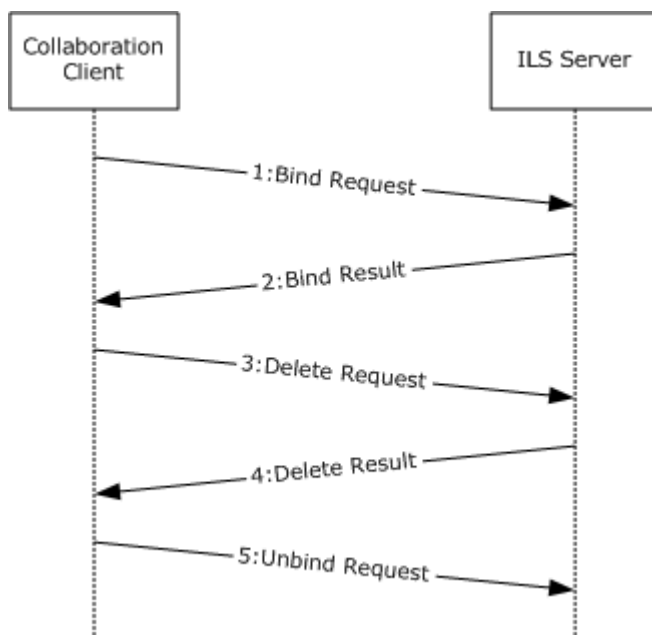
1. Bind Request: Client will "bind" to the ILS Server using the LDAP Bind Operation.

2. Bind Result of Bind Request from item 1.
3. Search Request: The client will then utilize the LDAP Search Operation to retrieve (query) a Time-To-Live (TTL) value. If the search operation is successful, the entry's **sttl** value will be reset to that value supplied in the search operation.
4. Search Result: If the search request is successful, then the *sttl* attribute will be returned.
5. Unbind Request: No results returned.

If any of the above LDAP operations fail OR an empty result set is returned to the Collaboration Client, it will be necessary for the Collaboration Client to re-register its entry on the ILS Server.

#### 4.4.5.5 Unregistering the Collaboration Client from the ILS Server

When the Collaboration Client has completed its communication session, it will unregister itself from the ILS Server. This is accomplished by issuing an LDAP Delete Operation. The DN used during the Registration phase will be supplied as the DN entry to delete. [<1>](#)



**Figure 11: Sequence diagram of Collaboration Client unregistering itself from the ILS Server**

1. Bind Request: Client will "bind" to the ILS Server utilizing the LDAP Bind Operation.
2. Bind Result.
3. Delete Request: The client will then utilize the LDAP Delete Operation to remove the entry from the ILS Server.
4. Delete Result.
5. Unbind Request: No results returned.

## 4.5 Task Security

There are no Task-specific security considerations. Please refer to the [Security](#) section of this specification and the Security sections of the referenced protocol Technical Documents.

## 5 Lookup Collaboration Client Task

This section describes the Lookup Collaboration Client Task. This Task is expected to be used by an individual whose goal is to locate another individual for the purpose of initiating a collaboration session with that individual.

### 5.1 Task Overview

#### 5.1.1 Task Purpose

The purpose of this Task is to allow a Collaboration Client such as NetMeeting to query ILS for the location information of other Collaboration Clients. As part of the query Task, the NetMeeting client will ask the ILS Server for a list of registered Collaboration Clients and then optionally choose to retrieve the location information for a specific member of the result set.

#### 5.1.2 Task Applicability

The reason a NetMeeting client would want to query an ILS Server would be to discover if there are any additional NetMeeting clients registered to participate in a collaboration session. The result of the query Task is the NetMeeting client receives the location information that can be used to then initiate a collaboration session between the two Collaboration Clients.

#### 5.1.3 Task Use Cases

##### 5.1.3.1 Stakeholders and Interests Summary

**Collaborative User 2:** Collaborative User 2 is a user of the system with the goal of registering a NetMeeting client with the ILS Server.

**Collaboration Client 2:** In order to participate in a collaboration session, the collaborative user must be running a Collaboration Client. In this use case, the client is NetMeeting.

**TAIL:** The interaction between Collaboration Client 2 and the ILS Server is done using the TAIL protocol ([\[MS-TAIL\]](#)).

**ILS Server:** The Internet Locator Service (ILS) is a name lookup service that provides registration services of Collaboration Clients in order to facilitate the lookup of a list of Collaboration Clients.

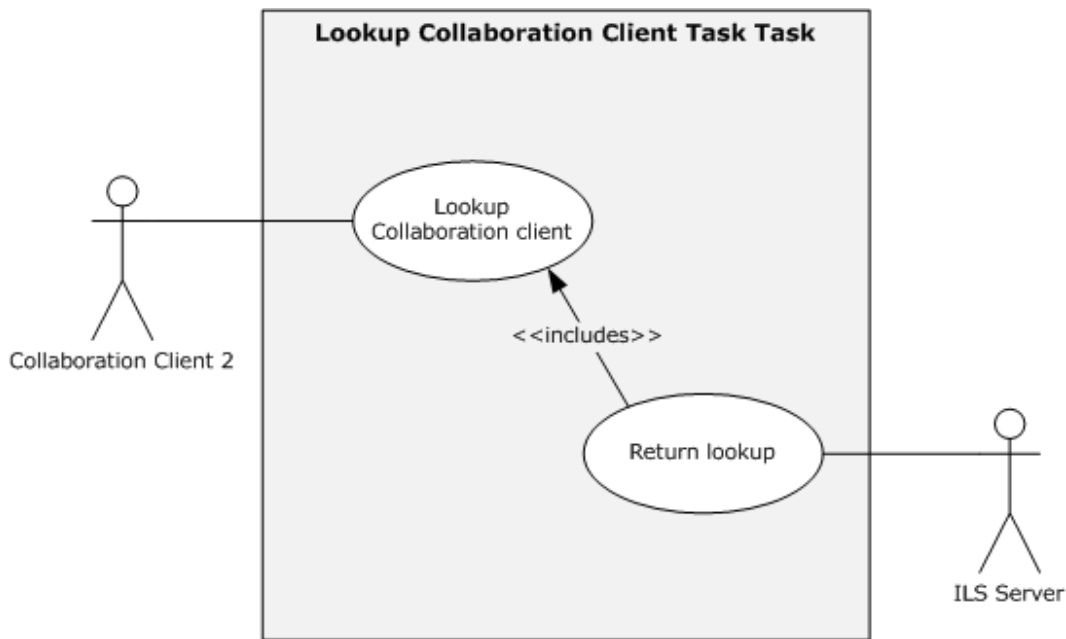
##### 5.1.3.2 Supporting Actors and Task Interests Summary

This Task is a standalone Task and is not an actor in any other Task.

##### 5.1.3.3 Use Case Diagrams

The following diagram provides an overview of how Collaborative User 2 performs a lookup of Collaborative User 1.





**Figure 12: UML diagram for the Lookup Collaboration Client Task**

#### 5.1.3.4 Collaboration Client Query - Client

**Goal:** To query an ILS Server for a list of registered Collaboration Clients in order to select one for which to receive its location information.

**Context of Use:** User wishes to find a Collaboration Client. Collaboration Clients are listed in the ILS.

**Direct Actor:** The direct actor is Collaboration Client 2. The Direct Actor's interest is initiating the query to the ILS Server and to receiving the location information for the selected registered Collaboration Client from the ILS Server.

**Primary Actor:** The primary actor is Collaborative User 2. The Primary Actor's interest is that the user is to discover the location information for an additional collaboration user that is currently registered with the ILS.

**Supporting Actor:** The TAIL protocol ([\[MS-TAIL\]](#)). TAIL is the protocol that is used to facilitate lookup services for an ILS Server.

**Preconditions:** A user has determined they want to collaborate with another user and will attempt to discover the other user's location information. The user is currently running a Collaboration Client on their computer and that client is configured to utilize an ILS. Collaboration Client 1 is currently registered with the ILS Server.

**Minimal Guarantees:** Collaboration Client 2 establishes a connection with the ILS Server, submits a query, and a result is returned.

**Success Guarantee:** Collaboration Client 2 receives the location information of another Collaboration Client that has published its availability to participate in a collaboration session.

**Trigger:** The Primary Actor has already started the Direct Actor application and uses the application user interface to query the ILS Server for a list of other registered collaboration clients.

Main Success Scenario:

1. Collaborative User 2 starts up Collaboration Client 2 on a client workstation.
2. Collaborative User 2 then uses Collaboration Client 2 to establish a session with an ILS Server to query the location information of additional Collaboration Clients that are currently registered with the ILS Server.
3. Collaboration Client 2 is returned a list of currently registered Collaboration Clients.
4. Collaborative User 2 selects a single entry from the list returned. The location information for the selected entry is then retrieved.
5. Collaboration Client 2 will then disconnect from the ILS.

Extensions: None.

## **5.2 Task Context**

This section describes the relationship between this Task and its environment.

### **5.2.1 Task Environment**

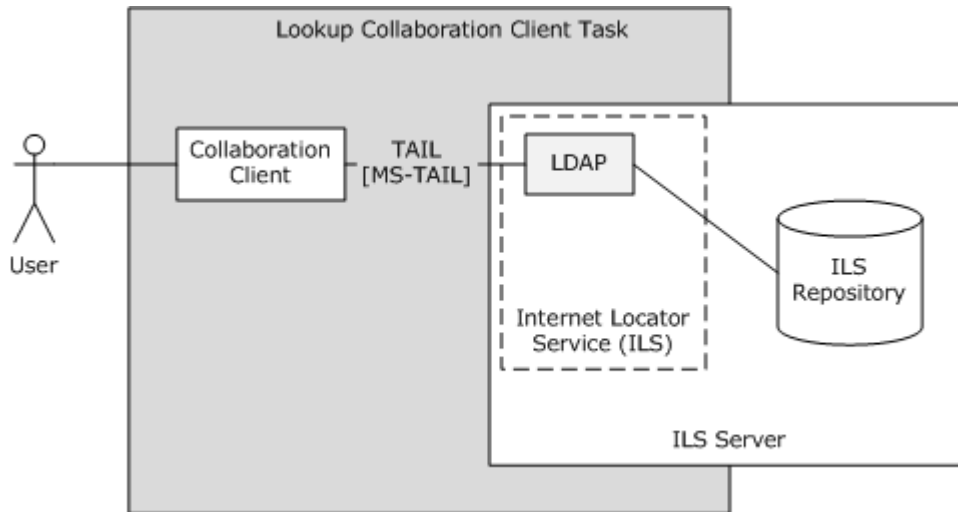
The typical environmental requirements for this Task are as follows:

- The Collaboration Client has network connectivity to an ILS Server.
- The Collaboration Client runs on a Windows XP or Windows 2003 operating system.
- The ILS Server is available over an IP network.
- The system requires a networked environment in which collaboration members are connected.
- The networked environment has to be capable of supporting TCP/IP traffic such as TCP and UDP.
- Firewall ports, network translation and network devices such as routers and switches are required to be configured in order that collaboration.
- Network traffic can flow between the Collaboration Client and the ILS Server.

### **5.2.2 Task Relationships**

The only entity this Task interacts with is an ILS Server.

### 5.2.2.1 Black-Box Relationship Diagrams - Locate Collaboration Client Task



**Figure 13: Black box diagram for the Lookup Collaboration Client**

### 5.2.2.2 Task Dependencies

The dependencies for this Task are as follows:

- Collaboration Client application software is NetMeeting version 3.0 which is included in Windows® XP operating system and Windows Server® 2003 operating system.
- ILS Server is an optional component of Microsoft Windows® 2000 Server operating system and Windows Server 2003.

### 5.2.2.3 Task Influences

There are no external influences for this Task.

### 5.2.3 Task Assumptions and Preconditions

The Collaboration Client does not automatically discover in any way the ILS Server. The location of the ILS Server MUST have been previously determined by the user and is manually configured within the Collaboration Client.

The ILS needs to be installation and network availability over TCP/IP is required.

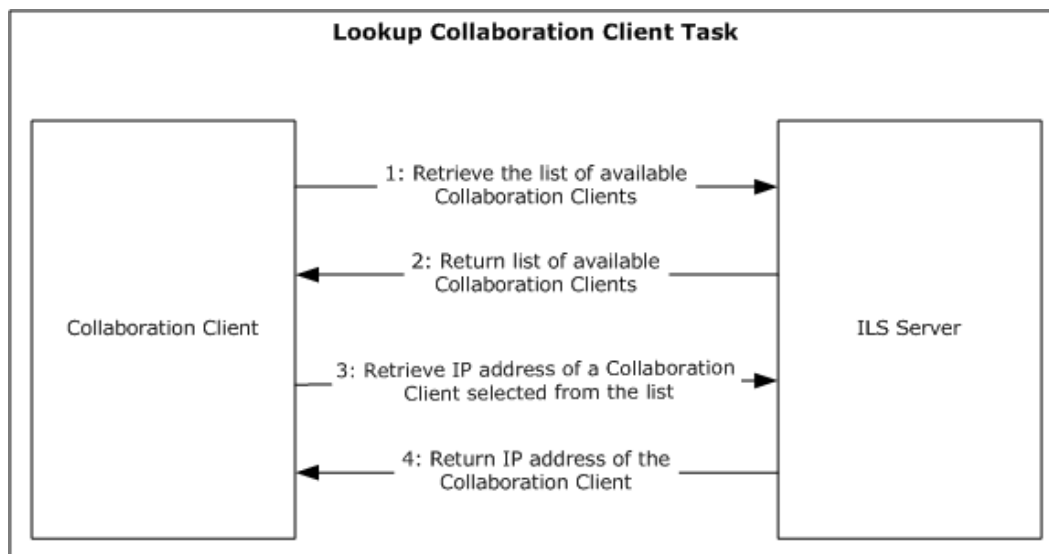
### 5.2.4 Task Versioning and Capability Negotiation

There is no versioning or capability negotiation of this Task.

## 5.3 Task Architecture

The following diagram illustrates the various components and relationships among them. The Collaboration Client requests the location of other collaboration clients from the ILS Server and the ILS Server will respond with a list of available collaboration clients. The Collaboration Client will then select one of the collaboration clients supplied in the list returned. Finally, the original Collaboration

Client retrieves the IP address of the selected client and establishes direct two-way communication using the address supplied.



**Figure 14: Component architecture for the Lookup Collaboration Client Task**

### 5.3.1 Task Architectural Constraints

There **MUST** be an ILS Server available that all Collaboration Clients are aware of. The various clients will use this ILS Server to register and locate other Collaboration Clients throughout the network. Each client must know the name/address of the ILS Server in advance.

### 5.3.2 Task Abstract Data Model

This section describes state established, used, and maintained by processing rules of this Task. State may be volatile or persisted. State may pertain to one, some, or all instances of the Task. The Task's state consists of the values of the named data elements (also called state variables) presented in this section. The overall organization of the data elements, with their names, is the Abstract Data Model. It is intended to facilitate the reader's conceptual understanding of the specification. While a Task's processing rules may depend upon associations established by the structure of its Abstract Data Model, such association can be achieved in other ways. Implementations may depart from this model so long as their external behavior remains consistent with that described in this document.

Each Collaboration Client that wants to be located by other collaboration clients will send information about itself to a specific ILS Server. The ILS Server in turn will store this information within its repository for fast data retrieval. Each new Collaboration Client wanting to initiate a conferencing session will retrieve the Collaboration Client's location information from the ILS Server. With the location information retrieved, the Collaboration Clients are then able to establish a direct connection between each other.

The clients will use the LDAP Extension ([\[RFC2589\]](#)) to retrieve the information from the ILS Server, typically over TCP port 1002 (default port).

### 5.3.3 Task Abstract Parameters

This section describes data passed to an instance of this Task at the time it is invoked or triggered. The parameters consist of the values of the named data elements presented in this section. The organization of a data element, with its names, is an Abstract Parameter. It is intended to facilitate the reader's conceptual understanding of the specification. While a Task's processing rules may depend upon associations established by the structure of its Abstract Parameters, such association can be achieved in other ways. Implementations may depart from this abstraction so long as their external behavior remains consistent with that described in this document.

When a client requests the location of available Collaboration Clients it sends an LDAP search request to the ILS Server. The information is transferred in LDAP packets and utilizes the LDAP Extension.

The following section describes the Collaboration Client information retrieved from the ILS Server.

#### 5.3.3.1 Client Retrieves Information from ILS Server

Client information can be retrieved from the ILS Server by using the LDAP Search Operation as found in [\[MS-TAIL\]](#) (section [4.3](#)).

The resulting IP address will then be utilized to make a direct connection to the client selected.

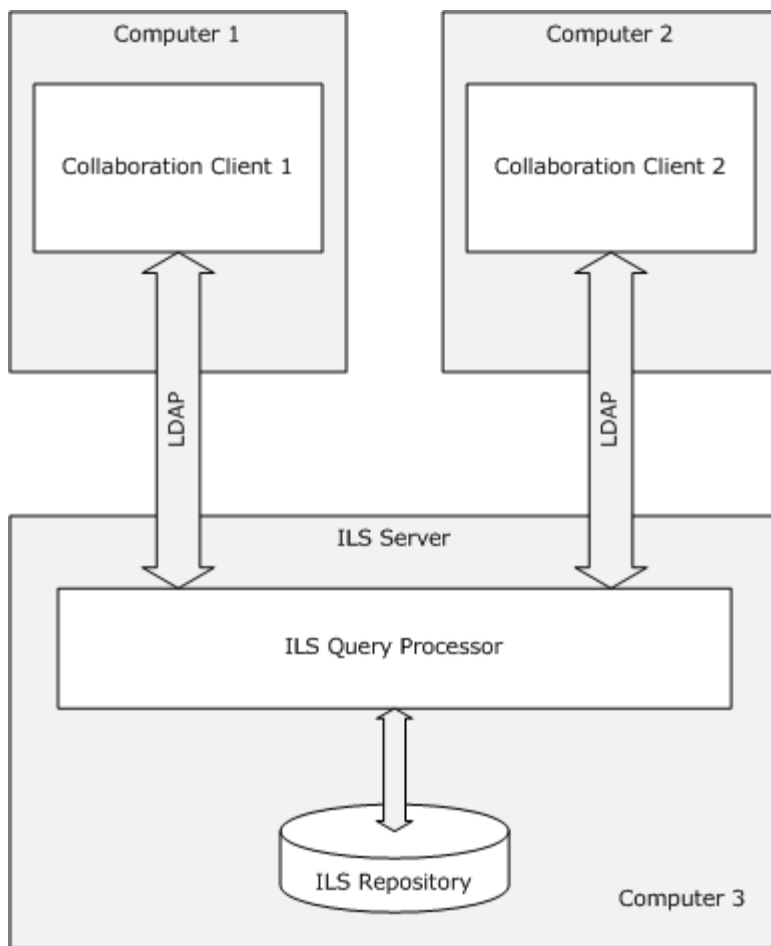
### 5.3.4 Task Abstract Results

This section describes data returned by an instance of this Task to its caller. The results consist of the values of the named data elements presented in this section. The organization of a data element, with its names, is an Abstract Result. It is intended to facilitate the reader's conceptual understanding of the specification. While a Task's processing rules may depend upon associations established by the structure of its Abstract Results, such association can be achieved in other ways. Implementations may depart from this abstraction so long as their external behavior remains consistent with that described in this document.

If the search is successful (see section [5.3.3.1](#)), then the standard LDAP return values will notify the client. Otherwise, an appropriate LDAP error will be returned.

### 5.3.5 White-Box Relationships

The following illustration outlines the various components present whenever ILS directory services are requested.



**Figure 15: White-box diagram for the Collaboration Client Registration Task**

### 5.3.6 Task Events

#### 5.3.6.1 Task Timers

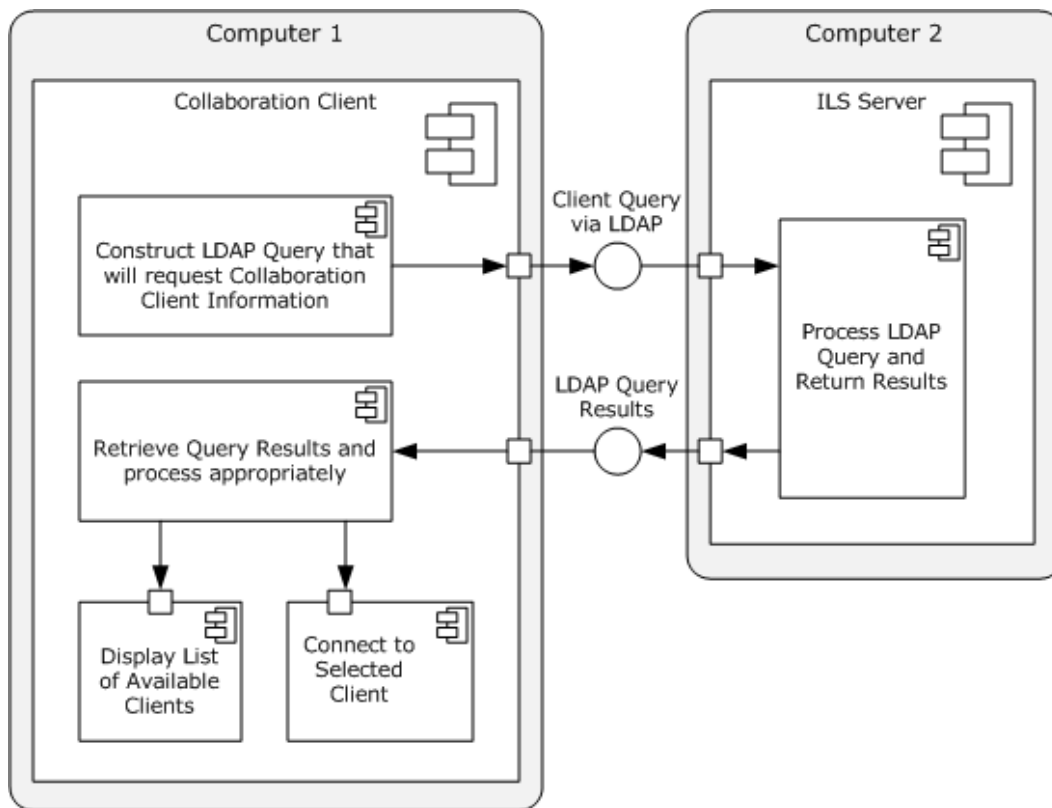
When a Collaboration Client requests the list of available Collaboration Clients from an ILS Server, it waits for a short period before discontinuing. An internal timer is utilized for this purpose.

#### 5.3.6.2 Task Non-Timer Events

There are no non-timer events.

### 5.3.7 Task Architecture and Communication

The follow diagram illustrates the various components required when looking up Collaboration Clients.

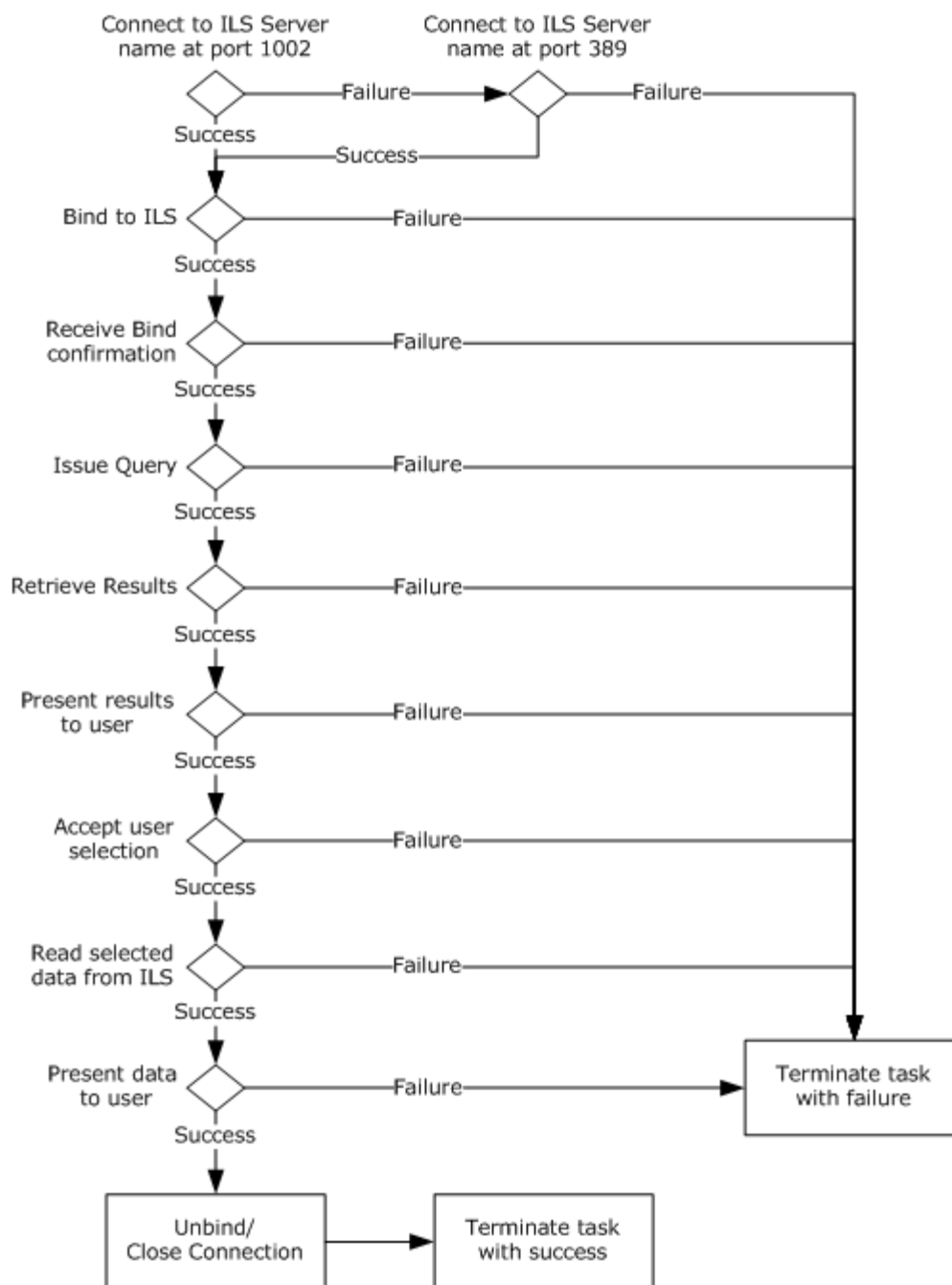


**Figure 16: Communication architecture diagram for the Lookup Collaboration Client Task**

Computer 1 will construct an LDAP query in order to retrieve a list of available Collaboration Clients previously registered with the ILS Server. The list of clients returned will then be presented to the user in some form of list display. The user will next select one of the clients from the list returned. This item will be utilized to construct a new query in order to retrieve additional information related to the selected client. The result will be returned and subsequently used to establish direct two-way communication with the target Collaboration Client.

### 5.3.8 Task Processing Rules

The following flowchart shows the operational flow for the Locate Collaboration Client Task.



**Figure 17: Processing rules flowchart for the Lookup Collaboration Client Task**

The abstract parameter, preconditions, and main success scenario of the Registration Task are described in the following sections.

#### Abstract Parameters

- Name of ILS Server (SERVER)



## Preconditions

None.

## Main Success Scenario

1. Bind to the ILS using LDAP.
  1. Resolve the name of the ILS Server to a network address if necessary. (The server location may have been given as a network address to the Collaboration Client.)
  2. Bind to the address, using port 1002. This establishes the LDAP connection.
  3. Receive Bind confirmation.
  4. Go to 2.
2. Issue Query to the ILS.
  1. Issue a Query request for all registered clients.
  2. Receive Query results.
  3. Go to 3.
3. Retrieve user-selected data from the ILS.
  1. Issue a Read command to retrieve data for selected entry.
  2. Receive data.
  3. Go to 4.
4. Unbind from the ILS.
  1. Issue an Unbind request to the ILS.
  2. Go to 5.
5. Close Connection.
6. Terminate Task with success.

## Extensions:

- 1.a.1 Unable to resolve name.
- 1.a1.a Terminate Task with error.
- 1.b.1 Unable to bind to address.
  - 1.b.1.a Retry using port 389.
    - 1.b.1.a.1 Unable to bind to address.
      - 1.b.1.a.1.a Terminate Task with error.
- 1.c.1 Did not receive response from server.
  - 1.c.1.a Terminate Task with error.

- 2.b.1 Did not receive results from server.
- 2.b.1.a Close connection.
- 2.b.2.b Terminate Task with error.
- 3.b.1 Did not receive results from server.
- 3.b.1.a Terminate Task with error.

### **5.3.9 Task Failure Scenarios**

#### **5.3.9.1 Failure Scenario: Failure to Bind**

In this scenario, Collaboration Client 2 fails to connect to the ILS.

Causes: This may be due to network failures, server failure, or a failure of the ILS to respond to the client's request.

Consequences: Failure of the Location Task. There are no consequences to the ILS.

Recognition: The failure is recognized by the Collaboration Client a) being unable to connect to the ILS at a network level, or b) failing to receive a Bind acknowledgement from the ILS.

Task Recovery: None.

Consequences of Recovery: None.

#### **5.3.9.2 Failure Scenario: Failure to Receive Results**

In this scenario, Collaboration Client 2 fails to receive results from the ILS.

Causes: This may be due to network failures, server failure, or a failure of the ILS to respond to the client's request.

Consequences: Failure of the Location Task. There are no consequences to the ILS.

Recognition: The failure is recognized by the Collaboration Client a) being unable to connect to the ILS at a network level after issuing a query, or b) failing to receive a result set from the ILS.

Task Recovery: None.

Consequences of Recovery: None.

### **5.4 Task Details**

This section contains the details that complete the descriptions in earlier sections of the document. These are needed to understand and implement this Task.

#### **5.4.1 Task Precondition Details**

The Collaboration Client does not automatically discover in any way the ILS Server. The location of the ILS Server is entered manually by the user within the Collaboration Client application. This is normally carried out upon application startup where some type of user dialog is presented. At that point, the user will input the name/address of the ILS Server to utilize during the registration process. This information will later be used in establishing a connection to the ILS Server.

## 5.4.2 Task Initialization of External Entities

The system requires a networked environment in which collaboration members are connected. The networked environment has to be capable of supporting TCP/IP traffic such as TCP and UDP. Firewall ports, network translation and network devices such as routers and switches are required to be configured in order that collaboration network traffic can flow between collaboration members.

## 5.4.3 Task Event Details

### 5.4.3.1 Task Timer Details

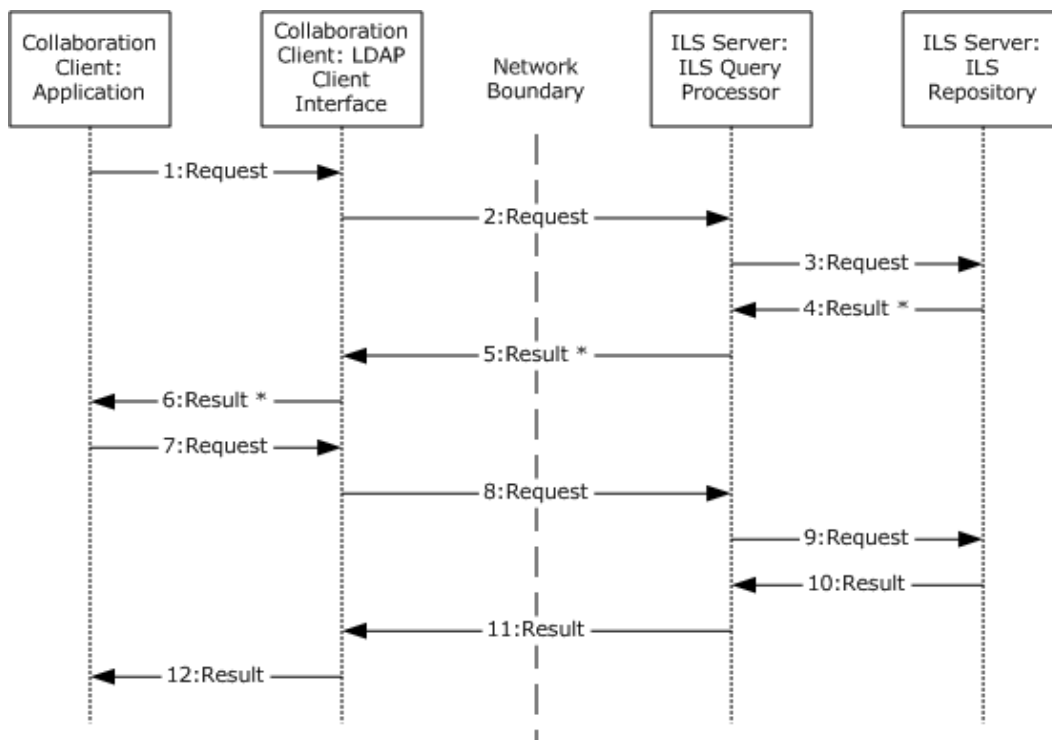
When a Collaboration Client requests the list of available Collaboration Clients from an ILS Server, it waits for 30 seconds before timing out. The timer is set using the LDAP Set\_Option Operation.

### 5.4.3.2 Task Non-Timer Event Details

There are no non-timer events.

## 5.4.4 Task Architectural Details

This section illustrates an example of a Collaboration Client retrieving registration information that other Collaboration Clients have placed on the ILS Server. The client will utilize several LDAP functions in order to accomplish the requests. See [\[LDAP\]](#).



**Figure 18: Sequence diagram for the Lookup Collaboration Client Task**

1. Request: The initial connect process is carried out using the LDAP Initialization Operation. The name of the target ILS Server is supplied at this point along with the TCP port to use (default port 1002). Next, the client will "bind" to the ILS Server utilizing the LDAP Bind Operation. When

these operations have successfully completed, the client will then utilize the LDAP Search Operation to retrieve (query) a list of available Collaboration Clients. (See section [5.3.3.1](#) for the required parameters and filters needed for the ILS query.)

2. Request: LDAP will send this information over the network to the destination ILS Server. The ILS Server will then pick up this information and validate it.
3. Request: The Collaboration Client registration data will then be retrieved from the "ILS Repository" based on the Collaboration Client's request.
4. Result: The result of the operation will be sent back to the client.
5. Result: The result of the operation will be sent back to the client.
6. The client will then review the results provided by the LDAP client operations and take appropriate action.

Note that many entries may be present in the ILS Repository. If a large result set is available, then the Collaboration Client and the ILS Server MAY use the PAGE\_RESULT operation to page through the results.

7. Request: When the list of available Collaboration Clients has been received, the client application will select one of the clients it desires to collaborate with. At this point, a new request (query) is sent to the ILS Server. The LDAP Search Operation is utilized for this purpose. The required data for the search operation is described in section [5.3.3.1](#).
8. Request: LDAP will send the request over the network to the destination ILS Server. The ILS Server will pick up the information and validate it.
9. Request: The requested information will be retrieved from the ILS Repository.
10. Result: The result of the operation will be sent back to the client.
11. Result: The result of the operation will be sent back to the client.
12. Result: The client will then review the results provided by the LDAP client functions and take appropriate action.

## **5.4.5 Task Processing Rule Details**

This section describes the processing rules required to look up Collaboration Clients registered on the ILS Server.

### **5.4.5.1 Initialization**

The first process is to initialize the subsystems used by the Collaboration Client application.

The client will need to open an LDAP session to the desired ILS Server. This is accomplished using the LDAP Open Operation with the specified HostName (or desired ILS Server) and TCP port 1002 (default port).

### **5.4.5.2 Binding to the ILS Server**

This operation is required to allow the client to "bind" itself to the ILS Server. The LDAP Bind Operation is used to accomplish this using anonymous authentication.

#### 5.4.5.3 Retrieve List of Available Collaboration Clients from the ILS Server

When the initialization phase is complete, the next step is to retrieve a list of available Collaboration Clients from the ILS Server (see section [5.3.3.1](#)). This is accomplished using the LDAP Search Operation. The DN is set to "objectClass=RtPerson" and the scope of the search should indicate that a search should be performed at the base-entry only. The search filter is set to "(&(objectClass=RtPerson)(cn=\*)(sappid=ms-netmeeting)(sprotid=h323))" and the attributes supplied will be those attributes defined in section [5.3.3.1](#).

#### 5.4.5.4 Retrieve Address of a Collaboration Client of Interest

When the list of available Collaboration Clients has been retrieved, it is generally desired to communicate with one of the active clients. This is accomplished by retrieving the IP address of that client. In order to accomplish this, use the LDAP Search Operation with the Distinguished Name set to "objectClass=RtPerson". The scope of the search should indicate that a search should be performed at the base-entry only. The search filter is set to "(&(objectClass=RtPerson)(cn=<email\_address>))" where <email\_address> is the Common Name of the person of interest (for example, "someone@example.com"), and the attribute requested will have the one entry of "sipaddress".

If the LDAP Search Operation was successful, the returned IP address can be retrieved using the appropriate LDAP data retrieval operations (for example, LDAP Result Operation, LDAP First\_Attribute Operation, LDAP Get\_Values Operation).

If an error occurs during any of these LDAP operations, then the LDAP Abandon Operation should be issued in order to abandon the operation requested.

After the retrieval attempt, the LDAP Unbind Operation should be issued.

### 5.5 Task Security

There are no Task-specific security considerations. Please refer to the Security section ([6](#)) of this specification and the Security sections of the referenced protocol Technical Documents.

## 6 Security

This section documents security issues common to all Tasks that are not otherwise described in the Technical Documents (TDs) for the protocols used in the Task. It does not duplicate what is already in the protocol TDs unless there is some unique aspect that applies to the system as a whole.

There are no additional security assumptions or pre-conditions required to register with and/or query an ILS Server. There is no shared domain registration or affinity between the Collaboration Client and the ILS Server required.

## 7 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Windows® 2000 operating system
- Windows® XP operating system
- Windows Server® 2003 operating system
- Windows Server® 2003 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

[<1> Section 4.4.5.5](#): The Windows 2000 implementation of ILS Server behaves slightly differently than specified in [\[RFC2589\]](#). For example, if the Collaboration Client requests a TTL value that is larger than the ILS Server allows, the TTL value may be set to a value smaller than the client requested.

## 8 Change Tracking

This section identifies changes that were made to the [MS-CSSO] protocol document between the May 2011 and June 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.



- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">1.2 References</a>	Added explanatory statement regarding the removal of the publishing year from Microsoft Open Specification document references.	N	Content updated.

## 9 Index

### C

[Change tracking](#) 48  
[Client registration - ILS](#) 11  
Collaboration Client Registration Task  
    [architecture](#) 16  
    [context](#) 14  
    [details](#) 27  
    [overview](#) 12  
    [security](#) 31  
    [subtasks](#) 12

### G

[Glossary](#) 7

### I

[ILS query](#) 11  
[Informative references](#) 8  
[Internet Locator Service](#) 10  
[Interoperability](#) 9  
[Introduction](#) 7

### L

[List of tasks](#) 9  
[Locating collaboration client](#) 10  
Lookup Collaboration Client Task  
    [architecture](#) 35  
    [context](#) 34  
    [details](#) 42  
    [overview](#) 32  
    [security](#) 45

### N

[Normative references](#) 8

### O

[Overview](#) 9

### P

Prerequisites  
    [client registration](#) 11  
    [ILS query](#) 11  
    [Internet Locator Service](#) 10  
    [locating collaboration client](#) 10  
[Prerequisites - overview](#) 10  
[Product behavior](#) 47

### R

References  
    [informative](#) 8  
    [normative](#) 8

[Required information](#) 10

### S

[Security](#) 46  
[Summary](#) 9

### T

Tasks  
    [Collaboration Client Registration](#) 12  
    [list of](#) 9  
    [Lookup Collaboration Client](#) 32  
    [Tracking changes](#) 48