www.novell.com/documentation

Trusted Applications

GroupWise® Software Developer Kit

November 2012

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Please refer to http://www.novell.com/info/exports/ (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell developer products, and to get updates, see the Novell Developer Web site (http://www.novell.com/developer). To access online documentation for Novell products, see the Novell Documentation Web site (http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

GroupWise® Trusted Application enables you to develop applications that can login to any user's mailbox without supplying the user's password and perform various tasks.

Currently, Trusted Application APIs are accessible through C++ only.

IMPORTANT: Unless otherwise marked, the features in GroupWise Trusted Application will work with GroupWise 7 and later versions.

This guide contains the following sections:

- Chapter 1, "Overview," on page 7
- Chapter 2, "Functions," on page 11
- Appendix A, "Revision History," on page 15

Audience

This guide is intended for GroupWise developers.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comment feature at the bottom of each page of the online documentation, or go to Novell Documentation Feedback (http://www.novell.com/documentation/feedback.html) and enter your comments there.

Additional Documentation

For additional GroupWise SDK documentation, see the Novell Developer Web site (http://www.novell.com/developer).

1 Overview

To become trusted, a third-party application must register with GroupWise when it is installed by using the GroupWise Trusted Application API. Each time an application is installed that uses this API, it creates a trusted application record for that installation and returns an application key. Once the application is authenticated with that key, GroupWise allows access to a user's mailbox as if it were the user.

Before you develop your trusted application, you should be familiar with the concepts in the following documents:

- ConsoleOne 1.3 User Guide (http://www.novell.com/documentation/lg/consol13/index.html)
- GroupWise Product Documentation (http://www.novell.com/documentation-index/index.jsp?category=GroupWise)

WARNING: Do not add your trusted application key to any log files or create a record of it that could possibly be compromised or create a security risk.

This section covers the following:

- Section 1.1, "Creating Your Installation Program," on page 7
- Section 1.2, "Installing a Trusted Application," on page 8
- Section 1.3, "Running a Trusted Application," on page 8
- Section 1.4, "Editing Properties of a Trusted Application Record," on page 8
- Section 1.5, "Securing the Trusted Application," on page 8
- Section 1.6, "Accessing a Trusted Application," on page 9
- Section 1.7, "Deleting a Trusted Application Record," on page 10

1.1 Creating Your Installation Program

To enable your application to become a trusted application, you must do the following in your installation:

1. Call CreateTrustedAppObject (page 12).
2. Include the header file (gwtapp.h) in the module that calls CreateTrustedAppObject.
3. Dynamically load gwtapp.dll or include it in your link file. You need to distribute gwtapp.dll file as part of your application.

The testapp.cpp file, located in the demo directory of the software download, demonstrates this process.

1.2 Installing a Trusted Application

To install a trusted application, you must have rights to access the domain database. Run the installation on the primary domain database.

If you install on a secondary domain, you create an unsafe record that replicates back to the primary domain. The primary domain then replicates a “safe” record back to any secondary domains. However, until a record is safe and has been replicated to all secondary domains, you cannot use that record. Also, if you install on a secondary domain and synchronization is not working between the secondary and primary domains, the trusted application won't be available until bidirectional communication is restored between the secondary and primary domains.

When a trusted application is installed, GroupWise generates a trusted application key. The key is an ASCII value returned in the szTrustedAppKey parameter when the installation program calls CreateTrustedAppObject. GroupWise also creates a trusted application record associated with that installation. This record contains a copy of the application key which is used to authenticate the application at runtime.

1.3 Running a Trusted Application

Each time the application is run, it passes its key to GroupWise. GroupWise then verifies the application has a record and that the key value matches the one in the record. If it does, GroupWise recognizes the application as a trusted application.

1.4 Editing Properties of a Trusted Application Record

To edit properties of a trusted application record, you must have system-level administration rights. The trusted application record is edited in ConsoleOne and the following properties can be modified:

- Description
- IP address
- IP port
- Requires SSL

1.5 Securing the Trusted Application

The szTCPAddress parameter of CreateTrustedAppObject (page 12) enables you to enter a TCP/IP address if you choose. That address then becomes the only address from which you can log in using that application record. After authenticating the key, GroupWise verifies the TCP/IP address and matches that address against the one in the trusted application record.

If you want to run your application from more than one machine, you can do either of the following:

- Leave the szTCPAddress parameter blank. The application can be run from any machine.
- Specify a unique application name and IP address with each installation. A trusted application record is created for each installation.

Secure Socket Layer (SSL) can be used to run the application on a secure wire. If you enable the bSSLRequired parameter, an SSL connection is required; and both the application and the Post Office Agent (POA) must be configured to use SSL.

1.6 Accessing a Trusted Application

Access to a trusted application is available through:

- Section 1.6.1, “GroupWise Object API,” on page 9
- Section 1.6.2, “IMAP and the GroupWise POA,” on page 9

1.6.1 GroupWise Object API

The GroupWise Object API has been extended to include a new trusted application method named SetTrustedApplicationCredentials.

Once an application supplies the trusted application name and key by calling this method, it can call the MultiLogin method to attach to any user's online mail box (trusted applications cannot automatically log in to a remote or caching mailbox.).

The vUserID and vCommandLine parameters already contain the appropriate commands and do not need to be modified. If a password is provided (using either the vPassword parameter or on the command line) it must be the correct password to access this box. If a password is not provided, the Object API attempts to log in to the user's box using the trusted application name and key. The GroupWise engine then verifies the trusted application credentials for this machine, including the trusted application name and key. If the trusted application credentials are correct, the application may access the user's database as if it were the user.

For example, if a trusted application was named “EncryptMessages” and was given a key of “1234567890,” the application should call the SetTrustedApplicationCredentials method (using C++ syntax) as follows:

```
gwApplication->SetTrustedApplicationCredentials
("EncryptMessages", "1234567890");
```

The application should then log in to user JDoe by calling MultiLogin. For example:

```
  gwApplication->MultiLogin("JDoe", "/ipa-199.99.99.99 /ipp-1677",
         NULL, egwNeverPrompt, NULL, &dispAccount);
```

If the application name and application key are found in the GroupWise post office database, the trusted application is allowed to connect to user JDoe's mailbox with full access to create messages, read items, etc.

1.6.2 IMAP and the GroupWise POA

The GroupWise IMAP implementation has been extended to include a new trusted application authentication mechanism, XGWTRUSTEDAPP. This mechanism is available if you see the string "AUTH=XGWTRUSTEDAPP" as part of the capability response from the GroupWise server, as follows:

```
C: A001 CAPABILITY
S: * CAPABILITY IMAP4rev1 AUTH=XGWTRUSTEDAPP
S: A001 OK CAPABILITY completed
```

When an application becomes a trusted application, the key needs to be provided as part of the challenge/response of the IMAP authenticate. The application issues the IMAP authenticate command, waits for the challenge response "+," and replies with the XGWTRUSTEDAPP command that includes the application name and key as a concatenated string (Base64 encoded). The two null-terminated strings should be combined (with null between them) and Base64 encoded.

```
C: A002 AUTHENTICATE XGWTRUSTEDAPP
S: +
C: XGWTRUSTEDAPP UHJvdG9jb2xfdGVzdAA2QkQwOTRDMTA5RjEwMD
                 AwQjExOThEMDA1NTAwMTYwMDZCRDA5NEMyMDlG
                 MTAwMDBCMTE5OEQwMDU1MDAxNjAw
S: A002 OK XGWTRUSTEDAPP authentication successful
```

The application is now considered an authenticated trusted application and can log in to any user's mailbox using the LOGIN command without supplying the user's password. While in the trusted application state, the LOGOUT token can be used to end perusal of the current user's mailbox. It can then be followed by another LOGIN without disconnecting and re-authenticating (or re-establishing the SSL connection).

1.7 Deleting a Trusted Application Record

To delete a trusted application record, you must do the following in your uninstall program:

1. Call DeleteTrustedAppObject (page 14).
2. Include the header file (gwtapp.h) in the module that calls DeleteTrustedAppObject.
3. Dynamically load gwtapp.dll or include it in your link file. (You need to distribute gwtapp.dll file as part of your application.)

The testapp.cpp file, located in the demo directory of the software download, demonstrates this process.

If DeleteTrustedAppObject is not called in your uninstall program, your administrator must use ConsoleOne to manually delete the trusted application record.

2 Functions

GroupWise Trusted Application contains the following functions:

- "CreateTrustedAppObject" on page 12
- "DeleteTrustedAppObject" on page 14

CreateTrustedAppObject

Creates a Trusted Application object.

Syntax

```
#include <gwtapp.h>

INT WINAPI CreateTrustedAppObject(
     char       *szDomainPath,
     char       *szAppName,
     char       *szAppDesc,
     char       *szTCPAddress,
     WORD        uwTCPPort,
     BOOL        bSSLRequired,
     BOOL        bRequiresQueueing,
     BOOL        bMessageRetention,
     BOOL        bOverwrite,
     char       *szTrustedAppKey);
```

Parameters

szDomainPath

(IN) Points to an ASCII path to the primary domain. Maximum length: 260 bytes.

szAppName

(IN) Points to the name of the trusted application. The application names must be unique when the application is installed more than once.

szAppDesc

(IN) Points to a description of the trusted application for display in ConsoleOne or other utility (optional).

szTCPAddress

(IN) Points to the IP Address of the of the machine where the trusted application can be run (optional). The IP address can be in DNS or dotted format.

uwTCPPort

(IN) Specifies IP Port of the machine where the trusted application can be run (optional).

bSSLRequired

(IN) Specifies TRUE or FALSE to indicate the trusted application connection requires SSL.

bRequiresQueueing

(IN) Is currently not implemented.

bMessageRetention

(IN) Specifies TRUE or FALSE to indicate whether the trusted application provides message retention services.

bOverwrite

(IN) Specifies true or false to indicate the trusted application should overwrite a trusted application record with the same application name.

szTrustedAppKey

(OUT) Points to the unique access key assigned to the trusted application. Returns a 64 byte key plus null.

Return Values

Value	Constant
0	SUCCESS
1	INIT_ERROR
2	CONNECT_ERROR
3	NEW_RECORD_ERROR
4	FIELD_ERROR
5	GET_RECORD_ERROR
6	NOT_UNIQUE_ERROR
7	OVERWRITE_ERROR
8	GET_RECORD_ID_ERROR
11	AUTHENTICATION_ERROR

DeleteTrustedAppObject

Deletes a trusted application object.

Syntax

```
#include <gwtapp.h>

INT WINAPI DeleteTrustedAppObject(
     char        *szDomainPath,
     char        *szAppName,
     char        *szTrustedAppKey);
```

Parameters

szDomainPath

(IN) Points to an ASCII path to the primary domain. Maximum length: 260 bytes.

szAppName

(IN) Points to the name of the trusted application to be deleted. This is the same name that was used in CreateTrustedAppObject (page 12).

szTrustedAppKey

(IN) Points to the unique access key returned in the CreateTrustedAppObject (page 12).

Return Values

Value	Constant
0	SUCCESS
1	INIT_ERROR
2	CONNECT_ERROR
3	NEW_RECORD_ERROR
4	FIELD_ERROR
5	GET_RECORD_ERROR
9	KEY_DOESNT_MATCH_ERROR
10	DELETE_ERROR

A Revision History

The following table lists changes made to the GroupWise Trusted Application API documentation:

Release	Changes
November 2012	Reviewed and updated for use with GroupWise 2012.
October 2006	Added a warning about adding trusted application keys to any files (see Chapter 1, “Overview,” on page 7).
March 2006	Transitioned to updated Novell® documentation style sheets.
October 2005	Transitioned to revised Novell documentation standards.
June 2004	Added a note to Section 1.2, “Installing a Trusted Application,” on page 8 about the consequences of installing on a secondary domain.
February 2004	Made minor changes to move documentation from Leading Edge.
October 2003	Added the bMessageRetention parameter and an authentication Return Value (11) to the CreateTrustedAppObject (page 12) function.
October 2003	Updated name of the former egwPromptNever enumeration to egwNeverPrompt.
February 2003	Made small changes to update the documentation and improve technical accuracy.
September 2002	Initial release.