

Command Reference Guide

*Sun™ ONE Integration Server
Secure Trading Agent*

Version 1.0

816-6369-10
February 2003

Copyright (c) 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

THIS SOFTWARE CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC. U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms.

This distribution may include materials developed by third parties.

Sun, Sun Microsystems, the Sun logo, Solaris, and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

Copyright (c) 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

CE LOGICIEL CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L'AUTORISATION EXPRESSE, ÉCRITE ET PRÉALABLE DE SUN MICROSYSTEMS, INC. Droits du gouvernement américain, utilisateurs gouvernementaux - logiciel commercial. Les utilisateurs gouvernementaux sont soumis au contrat de licence standard de Sun Microsystems, Inc., ainsi qu'aux dispositions en vigueur de la FAR [(Federal Acquisition Regulations) et des suppléments à celles-ci. Distribué par des licences qui en restreignent l'utilisation.

Cette distribution peut comprendre des composants développés par des tierces parties.

Sun, Sun Microsystems, le logo Sun, Solaris, et Java sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des États-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régi par la législation américaine en matière de contrôle des exportations ("U.S. Commerce Department's Table of Denial Orders" et la liste de ressortissants spécifiquement désignés ("U.S. Treasury Department of Specially Designated Nationals and Blocked Persons"), sont rigoureusement interdites.

This product includes Hypersonic SQL.

ORIGINAL LICENSE (a.k.a. "hypersonic_lic.txt")

For content, code, and products originally developed by Thomas Mueller and the Hypersonic SQL Group:

Copyright (c) 1995-2000 by the Hypersonic SQL Group. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes Hypersonic SQL."
- Products derived from this software may not be called "Hypersonic SQL" nor may "Hypersonic SQL" appear in their names without prior written permission of the Hypersonic SQL Group.
- Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes Hypersonic SQL."

This software is provided "as is" and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Hypersonic SQL Group or its contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption). However caused any on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.

For work added by the HSQL Development Group (a.k.a. hsqldb_lic.txt):

For content, code, and products subsequently developed by the hsqldb

Development Group:

Copyright (c) 2001-2002, The HSQL Development Group

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer, including earlier license statements (above) and comply with all above license conditions.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution, including earlier license statements (above) and comply with all above license conditions.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQldb.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

ebScript Command Reference	7
ebScript Command Summary	7
Command Access	9
ebScript Commands	10
AddLocalPartyID	10
AddUserRoles	11
CloseConversation	12
DeleteCertificate	12
DeleteCPA	13
DeleteKeystore	13
DeleteMessage	14
DeployCPA	14
Help	15
ImportCertificate	15
ImportCPA	16
ImportKeystore	16
ListActions	17
ListCertificates	17
ListConversations	18
ListCPAPartyIDs	18
ListCPAs	19
ListKeystores	20
ListLocalPartyIDs	20
ListMessageEvents	20
ListMessages	21
ListRolesByUser	22
ListServices	23
Login	23
Ping	24
Quit	24

RemoveAllUserRoles	24
RemoveLocalPartyID	25
RemoveUserRoles	25
Reply	26
RetrieveAllPayloads	27
RetrievePayloads	28
Send	28
UndeployCPA	30
WhoAmI	30

ebScript Command Reference

This reference is part of the documentation set for Sun™ Open Net Environment (Sun ONE) Integration Server, Secure Trading Agent. The reference provides documentation to the commands available from the Secure Trading Agent ebScript utility. For information on using ebScript, refer to the “Secure Trading Agent Administration” chapter of the *Secure Trading Agent User’s Guide*.

ebScript Command Summary

[Table 1](#) provides a summary of all ebScript commands.

For information on access to ebScript commands, refer to [“Command Access” on page 9](#).

For details on each command, refer to [“ebScript Commands” on page 10](#).

Table 1 ebScript Command Summary

Command	Access	Description.
AddLocalPartyID	Administrator	Defines a local party Id.
AddUserRoles	Administrator	Assigns one or more agreement roles to the specified user.
CloseConversation	All Users	Closes the specified conversation.
DeleteCertificate	Administrator	Deletes the specified certificate from the system.
DeleteCPA	Administrator	Deletes the specified agreement from the system. The agreement must be removed from service (undeployed) before it can be deleted.
DeleteKeystore	Administrator	Deletes the reference to the specified keystore from the system.

Table 1 ebScript Command Summary (Continued)

Command	Access	Description.
DeleteMessage	All Users	Deletes the specified message from the system. The conversation must be closed before the message can be deleted.
DeployCPA	Administrator	Deploys the specified agreement.
Help	All	Displays information about ebScript commands.
ImportCertificate	Administrator	Imports the specified file as a certificate with the specified keyname.
ImportCPA	Administrator	Imports the specified file as an agreement with the specified name.
ImportKeystore	Administrator	Imports the specified file as a keystore with the specified keyname.
ListActions	All Users	Lists the actions available in the specified agreement for a specified service.
ListCertificates	All Users	Displays a list of imported certificates.
ListConversations	All Users	Displays a list of conversations.
ListCPAPartyIDs	All Users	Displays a list of party Ids defined in the specified agreement.
ListCPAs	All Users	Displays a list of agreements known to the system.
ListKeystores	All Users	Displays a list of keynames which reference keystores imported into the system.
ListLocalPartyIDs	All Users	Displays a list of party Ids that can be used by the local application.
ListMessageEvents	All Users	Displays the event log for a message.
ListMessages	All Users	Displays a list of messages.
ListRolesByUser	All Users	Displays the list of roles that have been assigned to the specified user.
ListServices	All Users	Lists the services available in a specified agreement.
Login	All	Logs in the specified user.
Ping	All Users	Pings the endpoint associated with the specified agreement.
Quit	All	Exits the application.

Table 1 ebScript Command Summary (*Continued*)

Command	Access	Description.
RemoveAllUserRoles	Administrator	Unassigns all roles for the specified user.
RemoveLocalPartyID	Administrator	Removes the specified party Id from the list of party Ids that the local application can represent.
RemoveUserRoles	Administrator	Unassigns one or more specified agreement roles for the specified user.
Reply	All Users	Responds to an incoming message.
RetrieveAllPayloads	All Users	Retrieves the payloads for the specified action in a specified agreement, and writes the payloads to the specified directory.
RetrievePayloads	All Users	Retrieves the payloads from the specified message.
Send	All Users	Sends a message according to the specified agreement, service, and action.
UndeployCPA	Administrator	Removes the specified agreement from service so it cannot send or receive messages.
WhoAmI	All Users	Displays information about the user currently logged in.

Command Access

ebScript enforces the following access to commands:

- **Administrator**
Available only to Secure Trading Agent administrators. Administrators must be logged into ebScript to access these commands.
- **All Users**
Available to Secure Trading Agent administrators and regular users. Users must be logged into ebScript to access these commands.
- **All**
Available to all users. You do not have to be logged into ebScript to have access to these commands.

ebScript Commands

This section provides details on each ebScript command. For a summary description of the ebScript commands, refer to “[ebScript Command Summary](#)” on [page 7](#).

The listing for each command uses the typographic conventions described in [Table 2](#).

Table 2 Typographic Conventions

Style	Description
Bold	The name of the command and any options to the command
<i>Bold Italic</i>	Arguments that require a specific value. Unless indicated by the use of square brackets, arguments specified for a command are required to execute the command.
[Square Brackets]	Command options and values that are not required to execute the command.

AddLocalPartyID

AddLocalPartyID *party_id* [*party_id_type*]

Parameter	Description
<i>party_id</i>	The local party Id, which should match a party Id in a CPA.
<i>party_id_type</i>	The type of the local party Id, which should match the type of the specified party Id. If the party Id in the CPA does not specify a type, then do not specify <i>party_id_type</i> with this command. In this case the value for <i>party_id_type</i> defaults to “URI.”

Access Administrator

Adds the specified party Id to the Secure Trading Agent's list of local party Ids.

An ebXML agreement references party Ids for each of the parties to the agreement. The agreement can also specify a type for the party Id. The type of the party Id typically defines a namespace, such as a URN, for the value of the party Id. If the type is not specified, a type of URI is assumed.

When editing or importing an agreement, Secure Trading Agent must know which party to the agreement is the local party. Secure Trading Agent maintains a list of local party Ids that it uses to determine which party is the local party.

AddUserRoles

AddUserRoles *user_id* *role* [*role* . . .]

Parameter	Description
<i>user_id</i>	The system user Id for the user being assigned an agreement role.
<i>role</i>	An agreement role.

Access Administrator

Assigns one or more agreement roles to the specified user.

Although Secure Trading Agent allows you to assign agreement roles to users, this release of Secure Trading Agent does not enforce user roles when sending or receiving messages.

CloseConversation

CloseConversation *conversation_id*

Parameter	Description
conversation_id	Unique identifier for the conversation.

Access All users

Closes the specified conversation.

A conversation is the series of message exchanges between trading partners as they execute the business process defined by the actions in an ebXML agreement. When all the actions in the business process have been completed, you can safely close the conversation. Conversations that are closed are not listed with the **ListConversations** command.

If a conversation has been closed before all actions in the business process have been completed, use the Secure Trading Agent Communications Center to activate the conversation.

DeleteCertificate

DeleteCertificate *keyname*

Parameter	Description
keyname	Name used to reference a certificate specified in an agreement.

Access Administrator

Deletes the specified certificate from the system.

DeleteCPA

DeleteCPA *cpa_name*

Parameter	Description
cpa_name	Name used to reference an agreement that was created in or imported into Secure Trading Agent.

Access Administrator

Deletes the specified agreement from the system. You must first remove an agreement from service before you can delete it. You can use the [UndeployCPA](#) command to remove an agreement from service.

CAUTION Deleting an agreement removes all information about the agreement from the system. This information cannot be recovered.

DeleteKeystore

DeleteKeystore *keyname*

Parameter	Description
keyname	Name used to reference a keystore (private key).

Access Administrator

Deletes the reference to the specified keystore from the system.

Secure Trading Agent uses the keystore to digitally sign messages as specified in an agreement. A single keystore file can be referenced by different keynames within a Secure Trading Agent system.

CAUTION If you delete a reference to a keystore from the system, but still have an agreement deployed that uses that reference to the keystore, you will not be able to send signed messages for that agreement.

DeleteMessage

DeleteMessage *message_key*

DeleteMessage -id *message_id*

Parameter	Description
<i>message_key</i>	Message key for the message to be deleted.
<i>message_id</i>	Message Id for the message to be deleted.

Access All users

Deletes the specified message from the system. The conversation must be closed before the message can be deleted.

This command takes two forms, one using a message key and the other using a message Id. A message key is a unique local identifier for messages in a Secure Trading Agent system. A message Id is a globally unique identifier for a message.

CAUTION Deleting a message removes all information about the message from the system. This information cannot be recovered.

DeployCPA

DeployCPA *cpa_name*

Parameter	Description
<i>cpa_name</i>	Name used to reference an agreement that was created in or imported into Secure Trading Agent.

Access Administrator

Puts the specified agreement into service, enabling it to send or receive messages as specified in the agreement.

An agreement must first be created using the Secure Trading Agent Communications Center or imported into the Secure Trading Agent system before it can be deployed. When creating or importing agreements, you specify the name (unique within a Secure Trading Agent installation) that is used to reference the agreement.

Help

Help [*command*]

Parameter	Description
command	An ebScript command, or the minimum sequence of letters that specify a unique command.

Access All

Displays information about ebScript commands. If you do not specify a command, Help displays summary information on all available commands.

Help is displayed only for the commands for which you have access. For information on access to ebScript commands, refer to [“Command Access” on page 9](#).

ImportCertificate

ImportCertificate *keyname filename*

Parameter	Description
keyname	Name used to reference a certificate specified in an agreement.
filename	Path and filename to the certificate file.

Access Administrator

Imports the specified certificate file into the Secure Trading Agent system.

Certificates are specified in agreements to implement transport security and to sign messages. Before an agreement can be deployed, all certificates referenced in the agreement must first be imported into the Secure Trading Agent system.

ImportCPA

ImportCPA *filename cpa_name*

Parameter	Description
filename	Path and filename to a CPA document.
cpa_name	Name used to uniquely identify an agreement within a Secure Trading Agent installation.

Access Administrator

Imports the specified CPA document file into the Secure Trading Agent system.

When importing agreements, the name you specify must be unique within a Secure Trading Agent installation. This name is used to reference the agreement in other commands, such as **Send**, **ListActions**, **DeleteCPA**, and others.

ImportKeystore

ImportKeystore *keyname filename [password]*

Parameter	Description
keyname	Name used to reference a keystore (private key).
filename	Path and filename of the keystore file.
password	Password needed for access to the specified keystore file.

Access Administrator

Imports the keystore file with the specified keyname. If password is not specified, ebScript prompts you for the password before importing the keystore file.

Secure Trading Agent uses the keystore to digitally sign messages as specified in an agreement. A keystore file can be referenced by different keynames within a Secure Trading Agent system.

ListActions

ListActions [-local] <cpa_name> <service>

Parameter	Description
-local	List actions available for the local party only.
cpa_name	Name used to reference an agreement that was created in or imported into Secure Trading Agent.
service	Name of a service in the CPA.

Access All users

Lists the actions available in the specified CPA for a specified service. If you omit the local flag, this command lists actions for both the local party and the other party.

ListCertificates

ListCertificates

Access All users

Displays a list of certificates imported into the Secure Trading Agent system.

ListConversations

ListConversations [-sd *start_date*] [-ed *end_date*]

Parameter	Description
-sd <i>start_date</i>	Use <i>start_date</i> to specify the earliest conversation to include in the list.
-ed <i>end_date</i>	Use <i>end_date</i> to specify the latest conversation to include in the list.

Access All users

Displays a list of all (active and closed) conversations. If you specify a start date or end date, the specified date must in the proper format.

Use the following format to specify dates:

yyyy-m-d

yyyy: four digit year

m: one or two digit month

d: one or two digit day of the month.

For example, the following command lists all conversations between September 20, 2003 and October 21, 2003:

```
ListConversations -sd 2003-9-20 -ed 2003-10-21
```

ListCPAPartyIDs

ListCPAPartyIDs *cpa_name*

Parameter	Description
<i>cpa_name</i>	Name used to reference an agreement that was created in or imported into Secure Trading Agent.

Access All users

Displays a list of party Ids defined in the specified agreement.

ListCPAs

ListCPAs

Access All users

Displays a list of agreements known to the system.

This command lists the name of the agreement and the agreement status. The status of an agreement can be any of the following:

Table 3 Agreement Status

Status	Description
Deployed	The agreement has been put in service, and can be used to send and receive messages between trading partners. An agreement must first be of status Supported before it can be deployed.
Invalid	The agreement is incomplete and cannot be deployed. Typically, an invalid agreement can be edited so it is supported. A supported agreement can then be deployed.
Supported	The agreement contains all the information necessary to be deployed.
Template	An agreement that is a basis for creating new agreements using the Secure Trading Agent Agreement Editor.
Undeployed	An agreement that was taken out of service. When an agreement is undeployed, it can no longer be used as a basis for sending and receiving messages.
Valid	The agreement is complete but cannot be deployed because it uses features not supported by Secure Trading Agent.

ListKeystores

ListKeystores

Access All users

Displays a list of keynames that have been imported into the Secure Trading Agent system.

A keyname references a specific keystore file (private key). A keystore file can be referenced by different keynames within Secure Trading Agent. Thus a listing from this command can contain several keynames that reference the same keystore file.

ListLocalPartyIDs

ListLocalPartyIDs

Access All users

Displays a list of party Ids that are known as a local party Id to a Secure Trading Agent installation.

An ebXML agreement references party Ids for each of the parties to the agreement. Secure Trading Agent must recognize one of these party Ids as a local party before it can import or edit the agreement. For more information, see [AddLocalPartyID](#).

ListMessageEvents

ListMessageEvents *message_key*

Parameter	Description
<i>message_key</i>	List events for the message with the specified <i>message_key</i> .

ListMessageEvents -id *message_id*

Parameter	Description
-id <i>message_id</i>	List events for the message with the specified <i>message_id</i> .

Access All users

Displays the event log for a message. *message_key* is the local identifier for the message. *message_id* is the global identifier for the message.

When Secure Trading Agent sends a message, it generates the local message key and the global message Id. Use **ListMessages** to view the message keys and message identifiers for messages sent or received by a Secure Trading Agent installation.

ListMessages

ListMessages [-out] [-in] [-sd *start_date*] [-ed *end_date*]

Parameter	Description
-out	List outbound messages.
-in	List inbound messages.
-sd <i>start_date</i>	Use <i>start_date</i> to specify the earliest message to include in the list.
-ed <i>end_date</i>	Use <i>end_date</i> to specify the latest message to include in the list.

Access All users

Displays a list of messages sent or received by a Secure Trading Agent installation. If no options are specified, all messages are displayed.

This command displays the following information about each message:

- Message Key
- Message Id
- Status
- Party From
- Party To
- Timestamp
- Number of Payloads

Use the following format to specify dates:

yyyy-m-d

yyyy: four digit year

m: one or two digit month

d: one or two digit day of the month.

For example, the following command lists outbound messages sent or received between September 20, 2003 and October 21, 2003:

```
ListMessages -out -sd 2003-9-20 -ed 2003-10-21
```

ListRolesByUser

ListRolesByUser *user_id*

Parameter	Description
user_id	User Id for the user whose roles are to be displayed

Access All users

Displays the list of roles that have been assigned to the specified user.

Although Secure Trading Agent allows you to assign agreement roles to users, this release of Secure Trading Agent does not enforce user roles when sending or receiving messages.

ListServices

ListServices *cpa_name*

Parameter	Description
cpa_name	Name used to reference an agreement created in or imported into Secure Trading Agent.

Access All users

Lists the services available in the specified agreement.

In an ebXML agreement, actions are grouped under services. When using the Secure Trading Agent Agreement Editor to create actions for an agreement, all actions are created under the service `S1IS_STA_Service`. Agreements created from another source may use other services to group actions.

Login

Login [-i] *user_id* [*password*]

Parameter	Description
-i	Displays a dialog to prompt for a user Id and password.
user_id	System user Id of the user to be logged into Secure Trading Agent.
password	System password for the user specified by <i>user_id</i> .

Access All

Logs in the specified user. If a password is not specified, the user is prompted for the password.

The `-i` option opens a dialog that allows you to specify a user and password. This option is useful when automating a series of ebScript commands in a file. For information on automating ebScript commands in a file, refer to the chapter, "Secure Trading Agent Administration," in the *Secure Trading Agent User's Guide*.

Ping

Ping *cpa_name*

Parameter	Description
<i>cpa_name</i>	Name of the agreement specifying the endpoint to ping.

Access All users

Pings the endpoint associated with the specified agreement.

Use ping to determine if the endpoint specified by an agreement is alive, and thus able to receive and send messages.

Quit

Quit

Access All

Exits ebScript.

RemoveAllUserRoles

RemoveAllUserRoles *user_id*

Parameter	Description
<i>user_id</i>	System user Id for the user having roles removed.

Access Administrator

Removes all roles assigned to the specified user.

Although Secure Trading Agent allows you to assign agreement roles to users, this release of Secure Trading Agent does not enforce user roles when sending or receiving messages.

RemoveLocalPartyID

RemoveLocalPartyID *party_id*

Parameter	Description
party_id	Party Id for a party that might be specified in a CPA.

Access Administrator

Removes the specified party Id from the list of local party Ids for the Secure Trading Agent installation.

An ebXML agreement references party Ids for each of the parties to the agreement. Secure Trading Agent must recognize one of these party Ids as a local party before it can import or edit the agreement. For more information, see [AddLocalPartyID](#) and [ListLocalPartyIDs](#).

CAUTION You should be careful about removing a local party Id for an agreement imported or deployed into your system. Secure Trading Agent relies on local party Ids to determine the local party to an agreement.

RemoveUserRoles

RemoveUserRoles *user_id role [role . . .]*

Parameter	Description
user_id	System user Id for the user having roles removed.
role	An agreement role.

Access Administrator

Removes one or more specified agreement roles for the specified user.

Although Secure Trading Agent allows you to assign agreement roles to users, this release of Secure Trading Agent does not enforce user roles when sending or receiving messages.

Reply

Reply [-sync] message_key action [filename [filename. . .]],

Parameter	Description
-sync	Wait for a confirmation that the message was sent.
message_key	Local identifier of the message that is being responded to.
action	Name of the action that this message is performing.
filename	Path and file name of the file containing a payload for this message.

Access All users

Continue the conversation of the specified message by responding with the specified action. All documents specified in the action must be attached to the message, and should be of the MIME type specified in the action.

By default, messages are sent asynchronously (queued for sending). To verify that a queued message was sent, check the status of the message using [ListMessages](#).

Use the `-sync` option to override the default behavior and wait for a confirmation that the message was sent.

The Reply command provides an alternative to using the Send command to continue a conversation. For more information, refer to the [Send](#) command.

RetrieveAllPayloads

RetrieveAllPayloads [-i] *cpa_name service action directory*

Parameter	Description
-i	Display a dialog to navigate to a directory to save attached documents.
cpa_name	Name used to reference an agreement that was created in or imported into Secure Trading Agent.
service	Name of a service in the agreement specified by <i>cpa_name</i> .
action	Name of the action for which attached documents are to be retrieved.
directory	Destination directory for saving documents attached to messages for the specified <i>action</i> .

Access All users

Retrieves all attached documents for the specified action in an agreement, and writes the documents to the specified directory.

This command retrieves the documents for all conversations that have pending messages for the specified actions. Use **RetrievePayloads** to retrieve documents for an action in a specific conversation.

The -i option opens a dialog that allows you to navigate to a destination directory for attached documents. This option is useful when automating a series of ebScript commands in a file. For information on automating ebScript commands in a file, refer to the chapter, "Secure Trading Agent Administration," in the *Secure Trading Agent User's Guide*.

RetrievePayloads

RetrievePayloads [-force] *message_key directory*

Parameter	Description
-force	Retrieve payloads even if they have previously been retrieved.
message_key	The local identifier for the message whose attached documents are to be retrieved.
directory	Directory for saving attached documents.

Access All users

Retrieves documents attached to the specified message, and writes them to the specified directory.

Send

Send [-i] [-sync] [-conv *conv_name*] *cpa_name service action*
[*filename filename. . .*]

Send [-i] [-sync] [-name *new_conv_name*] *cpa_name service action*
[*filename filename. . .*]

Send [-i] [-sync] *cpa_name service action*
[*filename filename. . .*]

Parameter	Description
-i	Displays a dialog that allows you to specify the files to be sent as attachments to the message.
-sync	Wait for a confirmation that the message was sent
-conv <i>conv_name</i>	Continue the conversation with the name specified by <i>conv_name</i> .
-name <i>new_conv_name</i>	Create a new conversation with the name specified by <i>new_conv_name</i> .
<i>cpa_name</i>	Name of the agreement containing the service and action for message to be sent.

Parameter	Description
<code>service</code>	Name of the service in the specified CPA containing the action for the message to be sent.
<code>action</code>	Name of the action to be sent.
<code>file_name</code>	Path and file name for a document to be attached to the message.

Access All users

For the specified agreement, sends a message for the action in the specified service. This command has three basic forms—one for continuing an existing conversation and two for starting a new conversation.

- If `-conv` is specified, continues the conversation specified by `conv_name`.
- If `-name` is specified, starts a new conversation with the specified name.
- If neither `-name` or `-conv` is specified, starts a new conversation with a unique conversation name generated by the system.

All documents specified in the action must be attached to the message, and should be of the MIME type specified in the action.

By default, messages are sent asynchronously (queued for sending). To verify that a queued message was sent, check the status of the message using [ListMessages](#).

Use the `-sync` option to override the default behavior and wait for a confirmation that the message was sent.

The `-i` option opens a dialog that allows you to navigate to documents attached to the message. This option is useful when automating a series of ebScript commands in a file. For information on automating ebScript commands in a file, refer to the chapter, “Secure Trading Agent Administration,” in the *Secure Trading Agent User’s Guide*.

For an alternate way to continue a conversation, refer to the [Reply](#) command.

UndeployCPA

UndeployCPA *cpa_name*

Parameter	Description
cpa_name	Name used to reference an agreement that was created in or imported into Secure Trading Agent.

Access Administrator

Removes the specified agreement from service. You can no longer send and receive messages based on an agreement that has been undeployed.

WhoAmI

WhoAmI

Access All users

Displays information about the user currently logged in.