ECXpert Support for ebXML Messaging

Sun™ ONE Integration Server, B2B

Version 3.6.2

816-6726-10
April 2003

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.

Products covered by and information contained in this service manual are 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.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

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 et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE “EN L'ÉTAT” ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

Contents

List of Figures

List of Tables

List of Procedures

List of Code Examples

ECXpert Support for ebXML Messaging

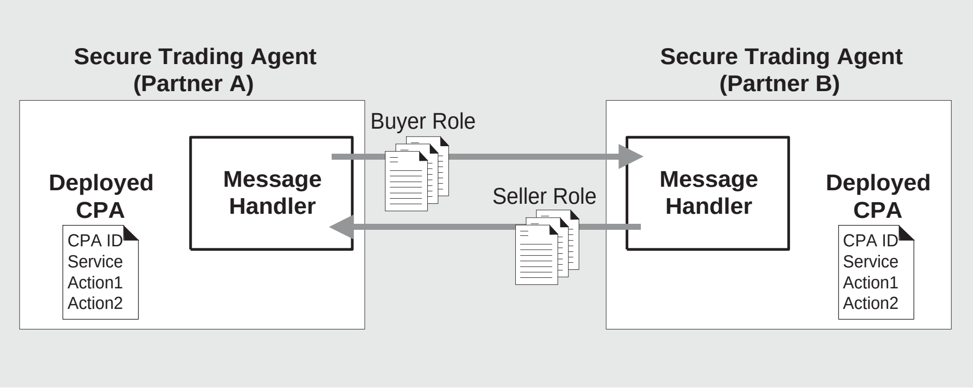
Sun™ Open Net Environment (Sun ONE) Integration Server, B2B Edition includes Sun ONE Integration Server, Secure Trading Agent. This document explains how to use ebXML Communications Agents provided with ECXpert to exchange ebXML messages.

About Secure Trading Agent

Secure Trading Agent is an implementation of ebXML specifications that allows you to securely and reliably exchange business documents between trading partners according to an agreement between the trading partners. This manual assumes that you have installed Secure Trading Agent and can successfully use it to deploy agreements and exchange messages with an ebXML trading partner. It also assumes you have a general understanding of ebXML specifications.

Figure 1 shows an ebXML messaging system between two Secure Trading Agent partners.

Figure 1 ebXML Messaging System Between Two ebXML Trading Partners



In Figure 1, both trading partners deploy identical copies of an *ebXML agreement* (Collaboration Protocol Agreement, or CPA). This agreement, identified by a unique *CPA ID*, defines one or more *actions* that each party can take. An action is typically a message with one or more attached business documents that one trading partner sends to the other trading partner. The agreement typically groups actions according to a *service* defined in the agreement.

An ebXML agreement also defines roles that specify who can perform actions specified in an agreement. A Secure Trading Agent administrator assigns roles to Secure Trading Agent users that authorize the users to perform the actions.

Figure 1 depicts Trading Partner A, acting in a buyer role, sending a message (defined by Action1) to Trading Partner B. Trading Partner B, acting in a seller role, responds by sending a message according to Action2. Each message consists of business documents specified for the action in the agreement.

NOTE When interoperating with ECXpert, Partner B defines a B2B User within Secure Trading Agent that is associated with the Seller Role. ECXpert uses the B2B User to retrieve messages from Partner B. For more information on using the B2B User to retrieve ebXML messages, refer to "ebXML Receive Communications Agent" on page 16.

Secure Trading Agent Documentation

For information on installing and using Secure Trading Agent, refer to the documentation provided with your copy of Secure Trading Agent. Of particular interest are the following manuals:

- *Secure Trading Agent Installation Guide*

  Provides instructions on installing Secure Trading Agent and using the registration process to verify that your installation is correctly configured.

- *Secure Trading Agent User's Guide*

  Provides an introductory chapter on ebXML principles and an administration chapter that shows how to configure Secure Trading Agent for your system. The other chapters show how to use the Secure Trading Agent components to deploy ebXML agreements and exchange messages with your trading partner.

- *Secure Trading Agent Tutorial*

  Provides a "hands on" introduction to authoring and deploying ebXML agreements and exchanging messages with your trading partner.

ebXML Communications Agents

ECXpert provides two Communications Agents, ebXML Receive and ebXML Send, for receiving and sending ebXML messages. Each Communications Agent interacts with a Secure Trading Agent installation, which must be installed on the same machine as ECXpert.

Figure 1 on page 14 depicts an ebXML messaging system between two ebXML trading partners, both using Secure Trading Agent. In reality, one trading partner in this collaboration may be using an ebXML messaging provider other than Secure Trading Agent.

ebXML Agreements

The ebXML agreement deployed by both partners in Figure 1 specifies information about the business collaboration, such as communication protocol, security level, MIME type of documents being sent, and other messaging parameters. However, ECXpert Communications Agents only need to know the following information to send and receive ebXML messages to and from a Secure Trading Agent installation:

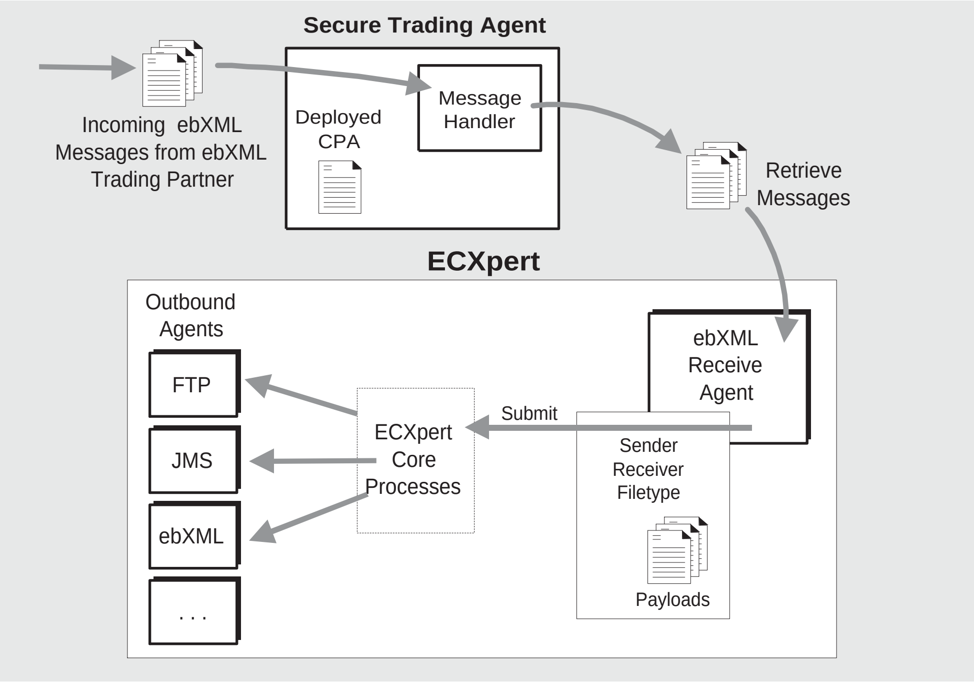
- Path to the Secure Trading Agent installation
- CPA ID of the deployed agreement
- Services defined by the agreement
- Actions defined by each service
- Role name associated with a Secure Trading Agent user

The following sections describe how the ebXML Send and ebXML Receive Communications Agents interact with Secure Trading Agent.

ebXML Receive Communications Agent

Figure 2 shows how an ebXML Receive Agent in ECXpert retrieves messages from a Secure Trading Agent trading partner. In Figure 2, ECXpert and Secure Trading Agent must be installed on the same machine.

Figure 2 ebXML Receive Communications Agent



In Figure 2, Secure Trading Agent represents an ebXML trading partner that is collaborating with another ebXML trading partner (not shown) according to a deployed ebXML agreement. The agreement specifies actions authorized for a B2B User role (for example, the agreement specifies actions for a *myB2BRole* role). The Secure Trading Agent installation depicted in Figure 2 has been configured to associate this role with a B2B User.

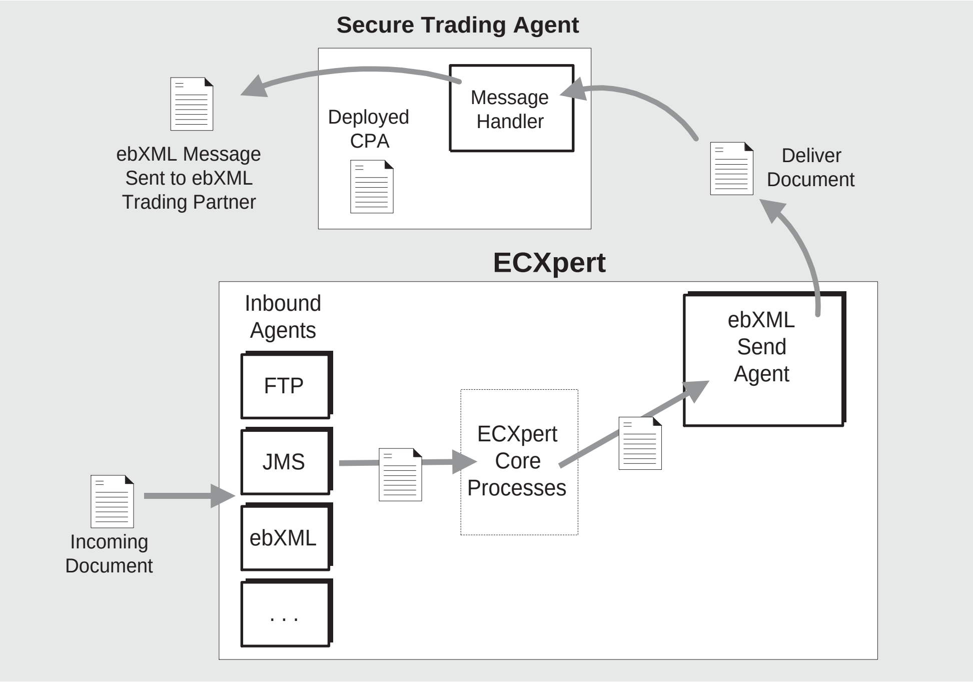
Figure 2 also shows an ebXML Receive Agent within ECXpert that has been configured to retrieve messages for the B2B User defined in Secure Trading Agent. The ebXML Receive Agent submits the documents from the retrieved messages according to sender, receiver, and filetype parameters. The filetype used for submission is based on configuration settings for the ebXML Receive Agent. (You typically specify these settings during installation of ECXpert.) The default filetype parameter specifies a CPA ID and action for the deployed CPA.

Once the documents are submitted, ECXpert performs any required parsing and translation of the documents and then routes them to the appropriate ECXpert outbound agent.

ebXML Send Communications Agent

Figure 3 shows how an ebXML Send Communications Agent in ECXpert sends documents to a Secure Trading Agent trading partner. In Figure 3, ECXpert and Secure Trading Agent must be installed on the same machine.

Figure 3 ebXML Send Communications Agent



In Figure 3, Secure Trading Agent represents an ebXML trading partner that is collaborating with another ebXML trading partner (not shown) according to a deployed ebXML agreement. In this figure, a document is submitted to ECXpert (either via ECXpert Inbound Communications Agents or directly). After any necessary parsing and translation, the document is routed to the ebXML Send Communications Agent. The ebXML Send Agent delivers the document to the Secure Trading Agent trading partner, which then sends the document as an ebXML message to its ebXML trading partner. The ebXML Send Agent specifies the CPA ID, service, and action to use when sending the ebXML message.

NOTE Unlike the ebXML Receive Agent, the ebXML Send Agent does not require a B2B User that is authorized for specific Secure Trading Agent roles.

Configuring ebXML Communications Agents

This section describes the configuration parameters for ebXML Communications Agents. You typically specify values for these parameters when you install ECXpert. If you do not specify these parameters at install time, or if you want to modify the values of these parameters, use the ECXpert administration interface as described in "Custom Configuration of ebXML Communications Agents" on page 52.

Table 1 lists the configuration parameters required at install time to configure ECXpert Communications Agents. This table also lists the actual parameter name specified in the ECXpert configuration file (`ecx.ini`).

Table 1 ebXML Communications Agents Parameters

Parameter	**Description**
EBROOT (`ebroot`)	The location of your Secure Trading Agent installation.
ebXML User (`ebxml_user`)	A system user that can log into Secure Trading Agent to retrieve messages. The ebXML User does not have to a Secure Trading Agent administrator, but must be a valid user on your system.
Password for ebXML User (`ebxml_passwd`)	System password for the ebXML user.
ebXML B2B User (`ebxml_b2b_user`)	The ebXML Receive Agent uses this parameter to retrieve messages from Secure Trading Agent. The ebXML B2B User corresponds to a user in Secure Trading that is associated with agreement roles in deployed ebXML agreements.
B2B Sending Member (`b2b_sender`)	Sender for an ECXpert partnership used for ebXML messaging.
Password for B2B Sending Member (`b2b_sender_passwd`)	The password for the B2B Sending Member. If you do not specify a password, then the sender should be trusted.
B2B Receiving Member (`b2b_receiver`)	Receiver for an ECXpert partnership used for ebXML messaging.
File Type Format (`filetype_format`)	The Receive Communications Agent uses the file type format to construct the filetype submission parameter when submitting retrieved messages to ECXpert for processing.

Setting Up Secure Trading Agent

Before you install ECXpert, it is recommended that you first install and configure Secure Trading Agent. This allows you to automatically configure the ebXML Communications Agents for ECXpert at install time. Otherwise, you have to customize the ECXpert configuration files to provide access to Secure Trading Agent. For information on customizing ECXpert configuration files to enable ebXML messaging, refer to "Custom Configuration of ebXML Communications Agents" on page 52.

ECXpert configuration requires the following information from your Secure Trading Agent installation:

Path to Secure Trading Agent The default paths are listed below. If you do not accept the default location when installing Secure Trading Agent, note the location you choose.

- Solaris™ platforms: `/opt/SUNWebXML/S1ISSTA`
- Windows platforms: `C:\Sun\S1ISSTA`

Secure Trading Agent User A valid user on your system that sends ebXML messages to and receives ebXML messages from an ebXML trading partner. The user can be the Secure Trading Agent administrator, but can also be a Secure Trading Agent regular user.

B2B User Name of a Secure Trading Agent user that must correspond to an *ebXML B2B User* that is defined for the ECXpert ebXML Receive Agent. This allows the ebXML Receive Agent to retrieve messages sent to the B2B User.

In Secure Trading Agent, you associate the B2B User with one or more roles specified in a deployed ebXML agreement. The roles correspond to receive actions specified in the agreement. The ECXpert ebXML Receive Agent retrieves messages associated with these receive actions

In Secure Trading Agent, use the `AddUserRoles` ebScript command to define the B2B User and associate the user with roles defined in an agreement. For example, to create a B2B User named *myB2BUser* and associate it with the role *Buyer* defined in an ebXML agreement, issue the following ebScript command:

```
AddUserRoles myB2BUser Buyer
```

NOTE Typically, `AddUserRoles` assigns ebXML agreement roles to valid users on your system. However, ECXpert does not require the B2B User to be a valid user on your system. The B2B User name should correspond to the ebXML B2B User you specify when you configure the ECXpert ebXML Receive Agent for ebXML messaging.

ebXML Agreement Information You need the following information from ebXML agreements deployed to Secure Trading Agent. ECXpert ebXML Retrieve Agents use some or all of this information to submit documents retrieved from Secure Trading Agent:

- **CPA ID**

  The unique identifier for an ebXML agreement. This identifier can be viewed in the Secure Trading Agent Agreement Editor before you deploy the agreement. Once an agreement is deployed, you can get the CPA ID by opening a text view of the agreement from the Secure Trading Agent Communications Center. The CPA ID is also available by examining an agreement file exported to a file system.

- **Service**

  ebXML agreements group actions by service. ebXML agreements created using the Secure Trading Agreement Editor have all actions grouped under a single service, `S1IS_STA_Service`. Agreements created with other editors can group actions under various services.

  To view the services in an agreement, from the Secure Trading Agent ebScript utility, issue the following command:

  `ListServices` *CPA_Name*

  CPA_Name is the name given to an ebXML agreement when it is created or imported into Secure Trading Agent.

- **Action**

  Each action in an agreement represents a message sent to or received from an ebXML trading partner. Actions typically contain one or more attached documents.

  To view the actions in an agreement, from ebScript issue the following command:

  `ListActions [-local]` *CPA_Name Service*

The optional `local` flag lists actions for the local trading partner only.

CPA_Name is the name given to an ebXML agreement when it is created or imported into Secure Trading Agent.

Service is the name of the service under which the actions are grouped.

- **ebXML Agreement Role**

  An ebXML agreement defines roles and associates the roles with actions defined in the agreement. ECXpert uses agreement roles as a basis for retrieving messages from Secure Trading Agent. Your ECXpert configuration for an ebXML Receive Agent defines an ebXML B2B User. The ebXML B2B User corresponds to a Secure Trading Agent user that has been associated with one or more agreement roles. This allows the ebXML Receive Agent to retrieve messages that have been associated with ebXML agreement roles.

Setting Up ECXpert for ebXML Messaging

This section describes how to set up ECXpert partnerships and service lists for sending and receiving ebXML documents. It assumes that you have installed Secure Trading Agent and properly configured ECXpert for ebXML messaging when you installed ECXpert.

Sending ebXML Messages

The procedure in this section shows how to set up a partnership to send documents as ebXML messages. It assumes that you have deployed an ebXML agreement in your Secure Trading Agent installation.

For each Send action in the deployed ebXML agreement, you must create or modify a partnership in ECXpert, specifying ebXML Send as the outgoing protocol.

➤ **To specify a partnership for an ebXML Send action**

1. From the Support interface, select Partnership.
2. Click Add to create a partnership or click Change to modify an existing partnership.

   Modify or edit the partnership as you normally do.
3. Specify ebXML Send for the Outgoing Protocol.
4. Specify the following information from the ebXML agreement deployed on the Secure Trading Agent installation with which you are communicating.

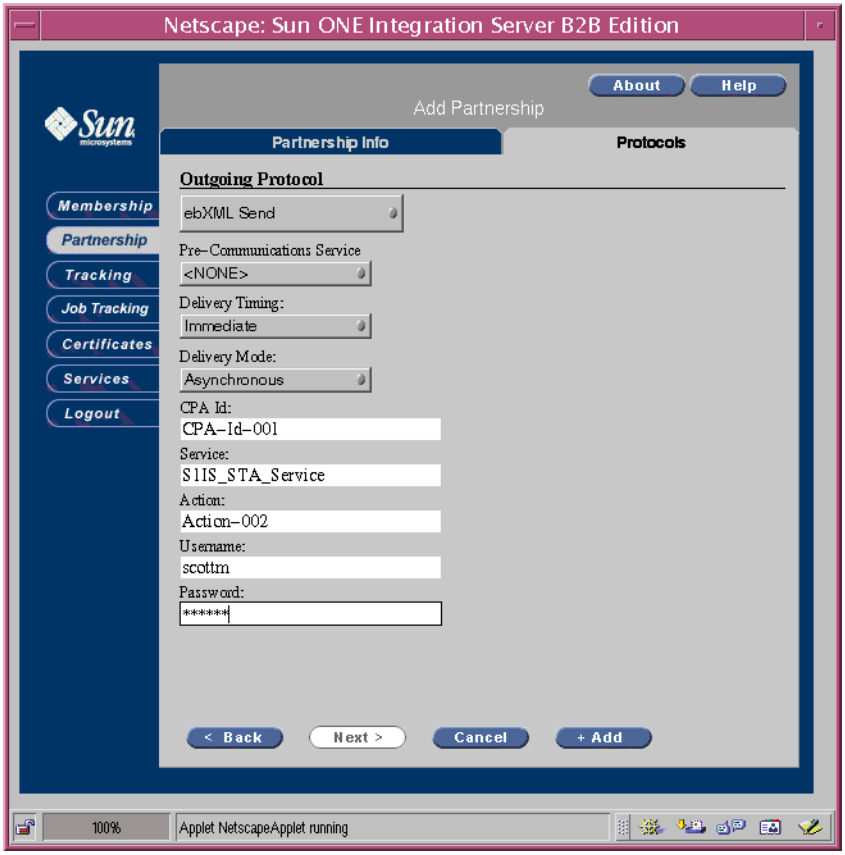
   CPA Id: The CPA ID of the deployed agreement containing the send action.

   Service: The service name in the agreement containing the action you want to perform.

   Action: The name of the action you want to perform.

   Username/Password: A valid Secure Trading Agent user name and password. This user is a valid system user in the environment in which you installed Secure Trading Agent.

This example uses the value *CPA-Id-001* for the CPA ID of your deployed agreement and assumes that the agreement contains a service *S1IS_STA_Service* with an action named *Action-002*.



Add Partnership

Partnership Info | Protocols

Outgoing Protocol

ebXML Send

Pre-Communications Service <NONE>

Delivery Timing: Immediate

Delivery Mode: Asynchronous

CPA Id: CPA-Id-001

Service: S1IS_STA_Service

Action: Action-002

Username: scottm

Password: ******

Receiving ebXML Messages

The procedures in this section show how to set up a partnership to receive documents that were sent as ebXML messages and how to create a Service List to handle documents received.

This section assumes that you have deployed an ebXML agreement in your Secure Trading Agent installation. It also assumes that within Secure Trading Agent, you have associated ebXML agreement roles with a B2B User. The Secure Trading Agent B2B User corresponds to the ebXML B2B User defined in your ECXpert configuration for the ebXML Receive Agent. ECXpert retrieves messages for each Receive action in the deployed agreement associated with a role for the B2B User.

For each Receive action in the deployed agreement for which you want to receive messages, you must create an ECXpert partnership. You must also create a Service List to handle the document type for ebXML messaging. This section assumes that you have configured the ECXpert ebXML Receive Agent using the default document type template for ebXML messaging. The default document type template is:

```
cpa-id;action-name
```

The following procedure for creating a partnership uses the value *CPA-Id-001* for the CPA ID of your deployed agreement and assumes that the agreement contains an action named *Action-001*.

➤ **To create a partnership to receive ebXML messages**

1. From the Support interface, select the Partnership.

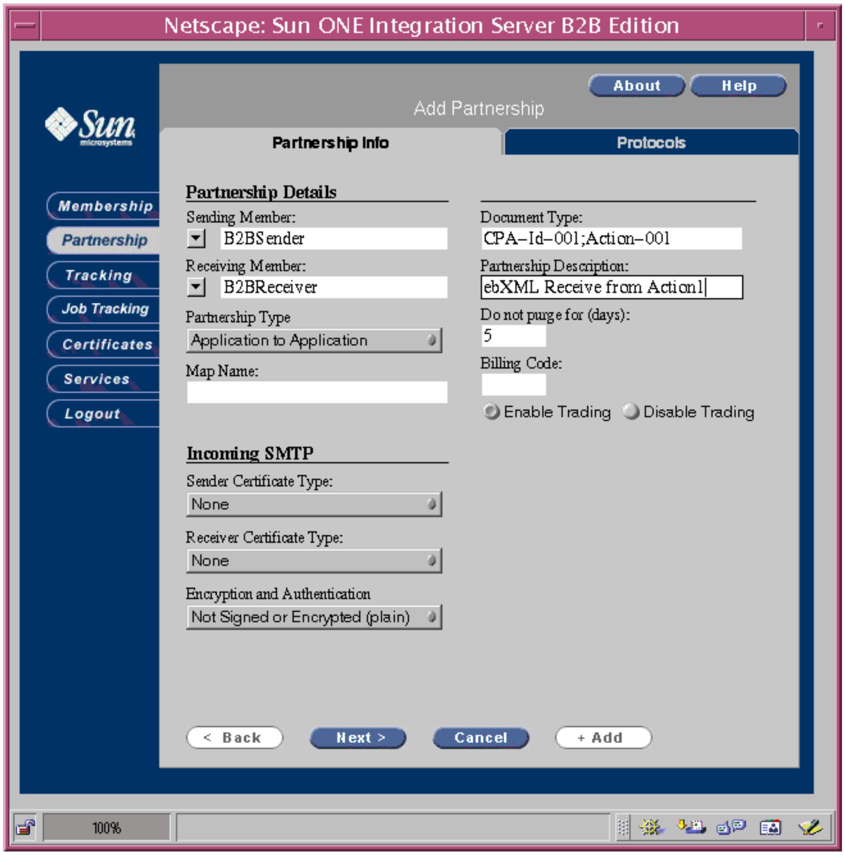
2. Select Add to add a new partnership.

3. Specify the partnership details as described below:

   Specify B2BSender and B2BReceiver as the sending and receiving members. These are the members you have already configured for the ebXML Receive Agent.

For the Document Type, specify values based on the Document Type template. For this example, you would specify the following for Document Type:

```
CPA-Id-001;Action-001
```



Partnership Details

Sending Member: B2BSender

Receiving Member: B2BReceiver

Partnership Type: Application to Application

Map Name:

Incoming SMTP

Sender Certificate Type: None

Receiver Certificate Type: None

Encryption and Authentication: Not Signed or Encrypted (plain)

Document Type: CPA-Id-001;Action-001

Partnership Description: ebXML Receive from Action1

Do not purge for (days): 5

Billing Code:

○ Enable Trading ○ Disable Trading

4. Select Add to add the partnership.

The following procedure for creating a Service List uses the value *CPA-Id-001* for the CPA ID of your deployed agreement and assumes that the agreement contains an action named *Action-001*.

➤ To create a Service List to handle ebXML messages

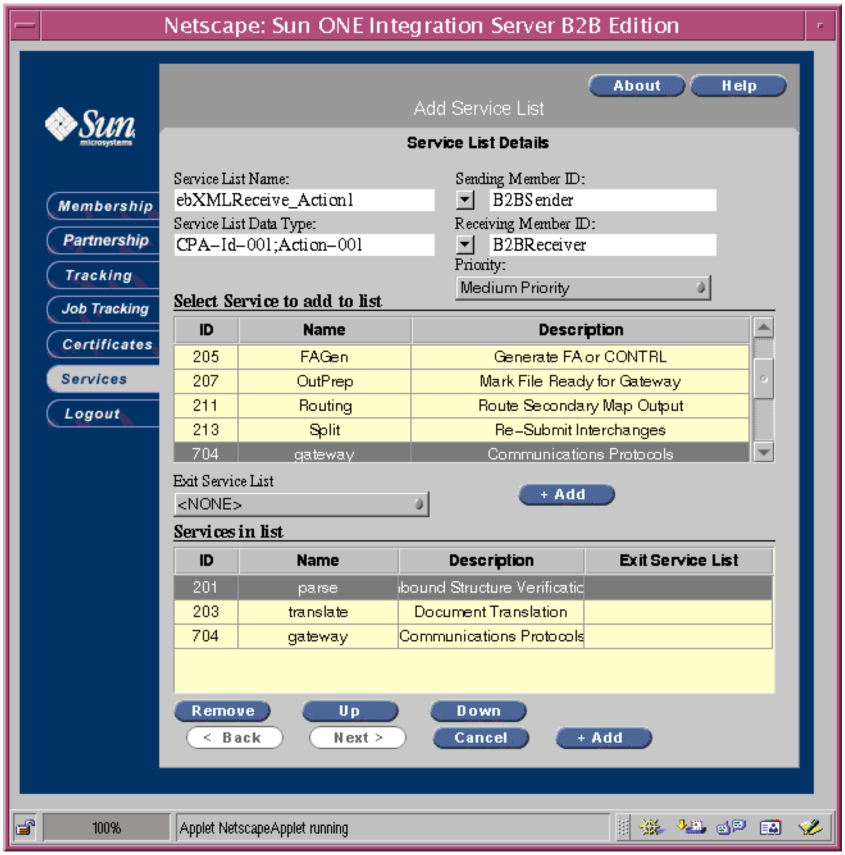
1. Create a new Service List.

2. For the Sending Member and the Receiving Member, specify the Sender and Receiver you specified for ebXML messaging when you installed ECXpert.

3. For the Service List Data Type, specify the values based on the Document Type template.

   For this example, you would specify:

   ```
   CPA-Id-001;Action-001
   ```



Add Service List

Service List Details

Service List Name: ebXMLReceive_Action1

Service List Data Type: CPA-Id-001;Action-001

Sending Member ID: B2BSender

Receiving Member ID: B2BReceiver

Priority: Medium Priority

Select Service to add to list

ID	Name	Description
205	FAGen	Generate FA or CONTRL
207	OutPrep	Mark File Ready for Gateway
211	Routing	Route Secondary Map Output
213	Split	Re-Submit Interchanges
704	gateway	Communications Protocols

Exit Service List: <NONE>

Services in list

ID	Name	Description	Exit Service List
201	parse	ıbound Structure Verificatic	
203	translate	Document Translation	
704	gateway	Communications Protocols	

4. Add services to the Service List, then click Add to add the Service List.

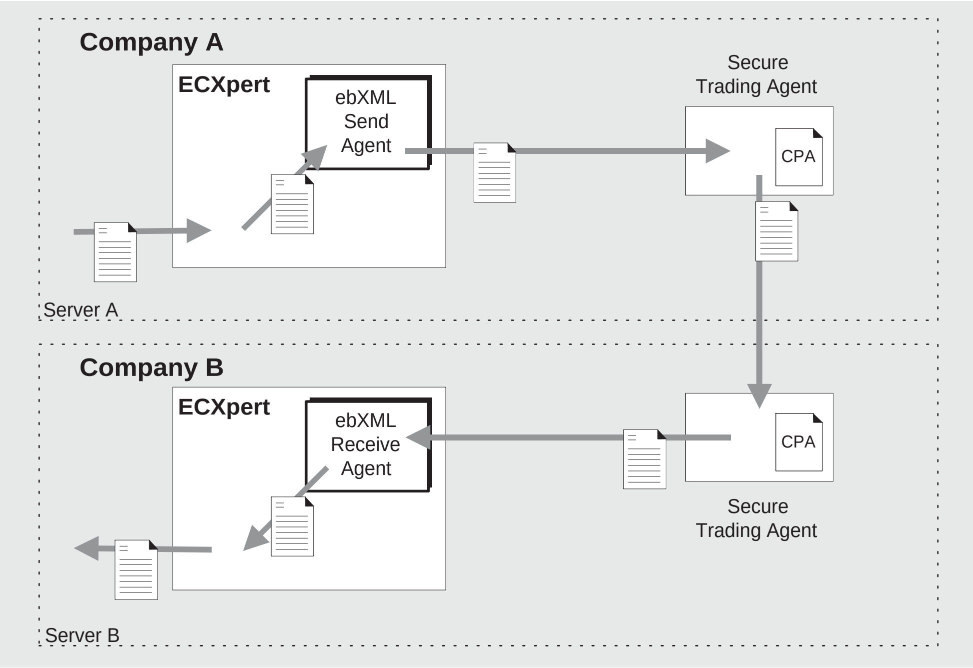
ebXML Messaging Example

This section shows how to use the sample ebXML agreement provided with your ECXpert distribution to send and receive ebXML messages using ECXpert. The example agreement specifies collaboration information between two trading partners, Company A and Company B. The agreement specifies that a user for Company A sends a message with an attached text document to Company B. It also specifies that Company B sends a message with a text document to Company A.

The ebXML messaging example shows how to use ECXpert to perform the Send action from Company A to Company B and also how to use ECXpert to retrieve the document sent to Company B. After Company B retrieves messages from Secure Trading Agent, it only submits documents to ECXpert that have been designated for roles associated with a B2B User.

Figure 4 shows the document flow for the ebXML messaging example. Figure 4 also shows that you must have separate machines for Company A and Company B. Each machine must have its own installation of ECXpert and Secure Trading Agent.

Figure 4 Document Flow for ebXML Messaging Example



Example Agreement

The example ebXML agreement, `ebxml_b2b.cpa`, can be found at the following location:

```
ECX_HOME/NS-apps/ECXpert/bin/ebxml/ebxml/test/ebxml_b2b.cpa
```

ECX_HOME is the location in which you installed ECXpert.

The example agreement specifies information for both Secure Trading Agent parties to exchange ebXML messages. For convenience, Table 2 lists specific information in the agreement that you use when configuring Secure Trading Agent and ECXpert to run the example.

Table 2 ebXML Agreement Information

CPA Information	Value
CPA ID	CompanyA-CompanyB-001
Party ID (Company A)	PartyA_ID
Endpoint (Company A)	http://*PartyA_Host*:8138/ebxml/msh (Replace *PartyA_Host* with the actual hostname you are using)
CPA Role (Company A)	PartyA_B2B_Role
Party ID (Company B)	PartyB_ID
Endpoint (Company B)	http://*PartyB_Host*:8138/ebxml/msh (Replace *PartyB_Host* with the actual hostname you are using)
CPA Role (Company B)	PartyB_B2B_Role
Service	S1IS_STA_Service
Action	A2B_SendAction

Before proceeding with the example, you must update the ebXML agreement to provide the correct endpoints for your Secure Trading Agent two installations.

➤ **To update the endpoints in the sample agreement**

1. Open the agreement, `ebxml_b2b.cpa`, in any text editor.

   The agreement is an XML file that can be edited using any text editor or XML editor. The modifications in this procedure can easily be handled by a plain text editor.

2. Within the agreement, search for “http://PartyA_Host:8138/ebxml/msh” and replace “PartyA_Host:8138” with the actual hostname and port you are using for Company A.

   The hostname is the machine on which you installed Secure Trading Agent. The default port is 8138. If you configured Secure Trading Agent to use a different port, be sure to specify this port in the ebXML agreement.

3. Within the agreement, search for “http://PartyB_Host:8138/ebxml/msh” and replace “PartyB_Host:8138” with the actual hostname and port you are using for Company B.

   These correspond to the edits made for Company A in Step 2 above.

4. Save the agreement.

   The agreement is now ready to be imported and deployed into your Secure Trading Agent installations for Company A and Company B.

Secure Trading Agent Set Up

The example requires you to install Secure Trading Agent on two systems, one for the Company A (Sender) and one for Company B (Receiver). After installing Secure Trading Agent, use ebScript to import and deploy the example on both systems. For Company B, you must also define the B2Buser and its associated role. You must login into ebScript as the Secure Trading Agent administrator to issue these commands.

Code Example 1 lists the ebScript commands for Company A.

Code Example 1 ebScript commands for Company A (Sender)

```
AddLocalPartyId  PartyA_ID
ImportCPA <PathToExample>/ebxml_b2b.cpa ebxml_b2b_CPA
DeployCPA ebxml_b2b_CPA
```

Code Example 2 lists the ebScript commands Company B. The `AddUserRoles` command listed in the example specifies a user, `ebxmlb2buser`, and associates the user with the agreement role, `PartyB_B2B_Role`. `ebxmlb2buser` should match the ebXML B2B User you have configured for Company B’s ebXML Receive Agent.

Code Example 2 ebScript commands for Company B (Receiver)

```
AddLocalPartyId  PartyB_ID
ImportCPA <PathToExample>/ebxml_b2b.cpa ebxml_b2b_CPA
DeployCPA ebxml_b2b_CPA
AddUserRoles ebxmlb2buser PartyB_B2B_Role
```

ECXpert Set Up for Company A

This section shows you how to set up a partnership for Company A to send ebXML messages. It contains procedures for the following tasks:

- Creating memberships for *ebXMLPartyA* and *ebXMLPartyB*
- Creating a partnership between ebXMLPartyA and ebXMLPartyB
- Creating a service list to route incoming documents to the ebXML Send Agent

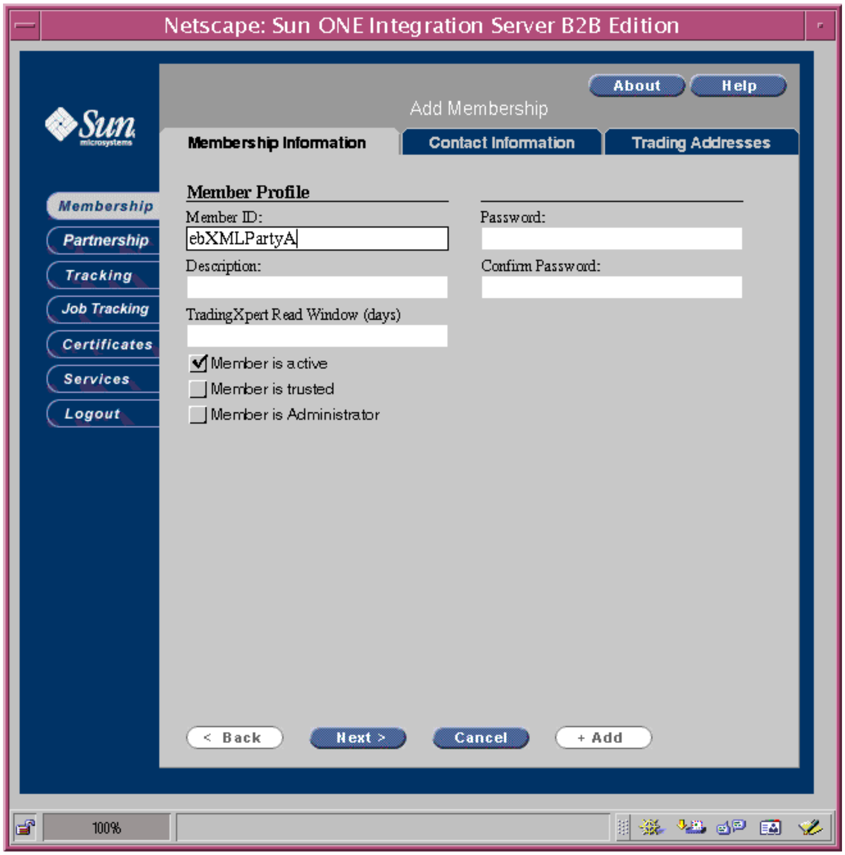
These procedures should be performed using the ECXpert installation on the machine you are using for Company A (Sender).

➤ To create memberships

1. From the Support interface, select Membership.
2. Select Add and specify the following:

Field	Value
Member ID	`ebXMLPartyA`
Select	`Member is Active`
Password	

If you do not specify a password, then the member should be trusted.



Member Profile

Member ID: ebXMLPartyA

Password:

Description:

Confirm Password:

TradingXpert Read Window (days)

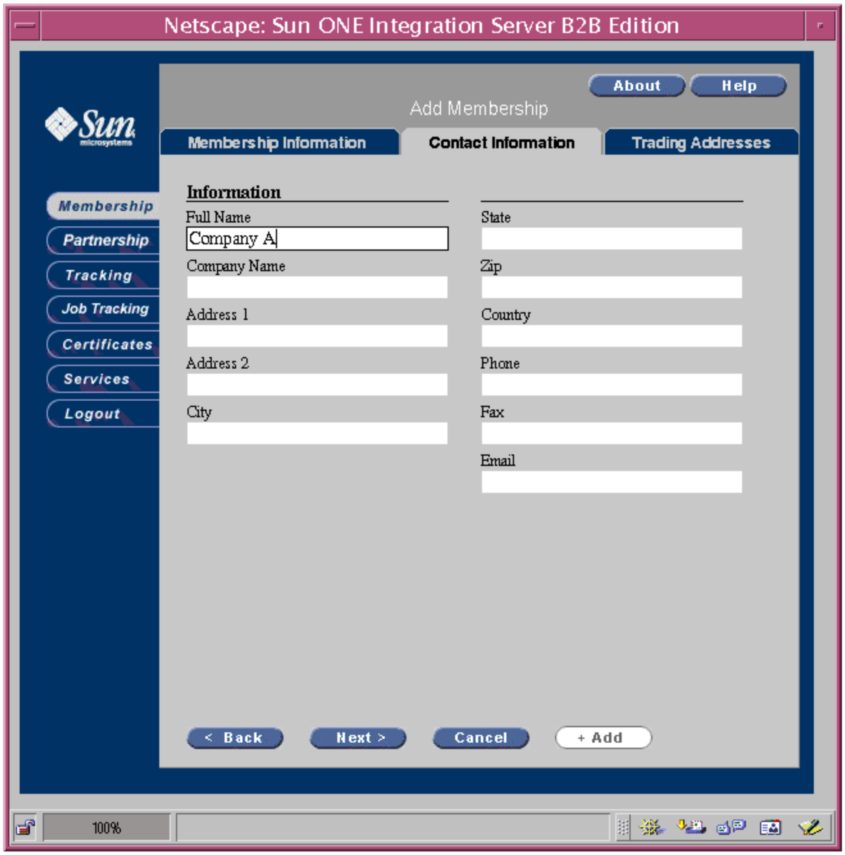
☑ Member is active

□ Member is trusted

□ Member is Administrator

3. Select Next to specify Contact Information.

   Specify the contact information as you like. The example is not dependent on these fields.



Information

Full Name: Company A

Company Name

Address 1

Address 2

City

State

Zip

Country

Phone

Fax

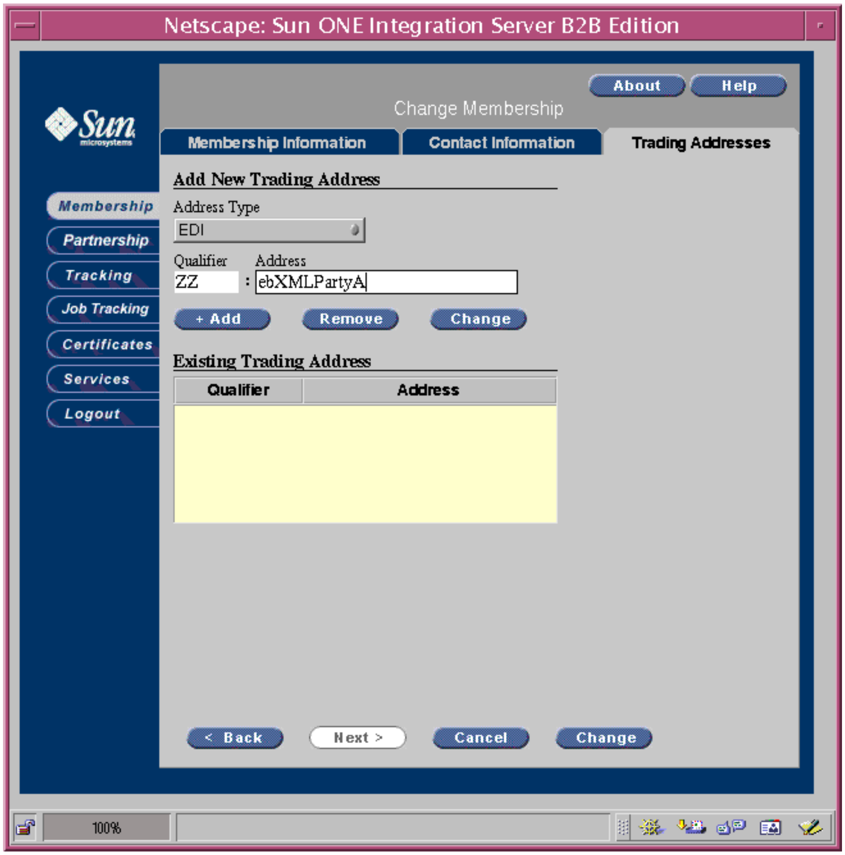
Email

4. Select Next to specify Trading Addresses.

   Specify an address type of `EDI`.

Add an address with the following values:

```
ZZ ebXMLPartyA
```



Netscape: Sun ONE Integration Server B2B Edition

Change Membership

Add New Trading Address

Address Type: EDI

Qualifier: ZZ

Address: ebXMLPartyA

Existing Trading Address

Qualifier	Address

5. Click Change to modify the member with the new trading address.

6. Repeat Step 1 through Step 5 to add member ebXMLPartyB, with the following differences:

   a. In Contact Information, specify `CompanyB` for the Full Name

   b. In Trading Addresses, specify the following address of type `EDI`:

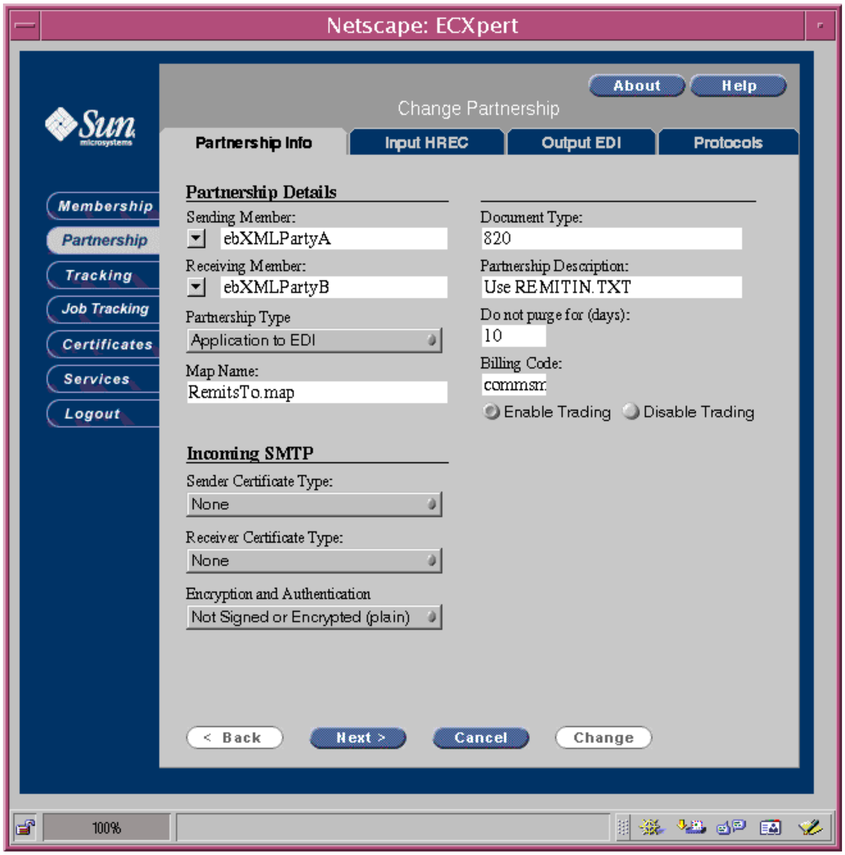
      ```
      ZZ ebXMLPartyB
      ```

7. Click Add to add the member.

➤ To create a partnership between ebXMLPartyA and ebXMLPartyB

1. From the Support interface, select Partnership.
2. Select Add and specify the following:

Field	Value
Sending Member	`ebXMLPartyA`
Receiving Member	`ebXMLPartyB`
Partnership Type	`Application to EDI`
Map Name	`RemitsTo.map`
Document Type	`820`
Partnership Description	`Use REMITIN.TXT`
Do not purge	`10`
Billing Code	`commsm`
Enable Trading	`On`
Incoming SMTP	`None, None, plain`



Change Partnership

Partnership Details

Sending Member: ebXMLPartyA

Receiving Member: ebXMLPartyB

Partnership Type: Application to EDI

Map Name: RemitsTo.map

Document Type: 820

Partnership Description: Use REMITIN.TXT

Do not purge for (days): 10

Billing Code: commsm

● Enable Trading ○ Disable Trading

Incoming SMTP

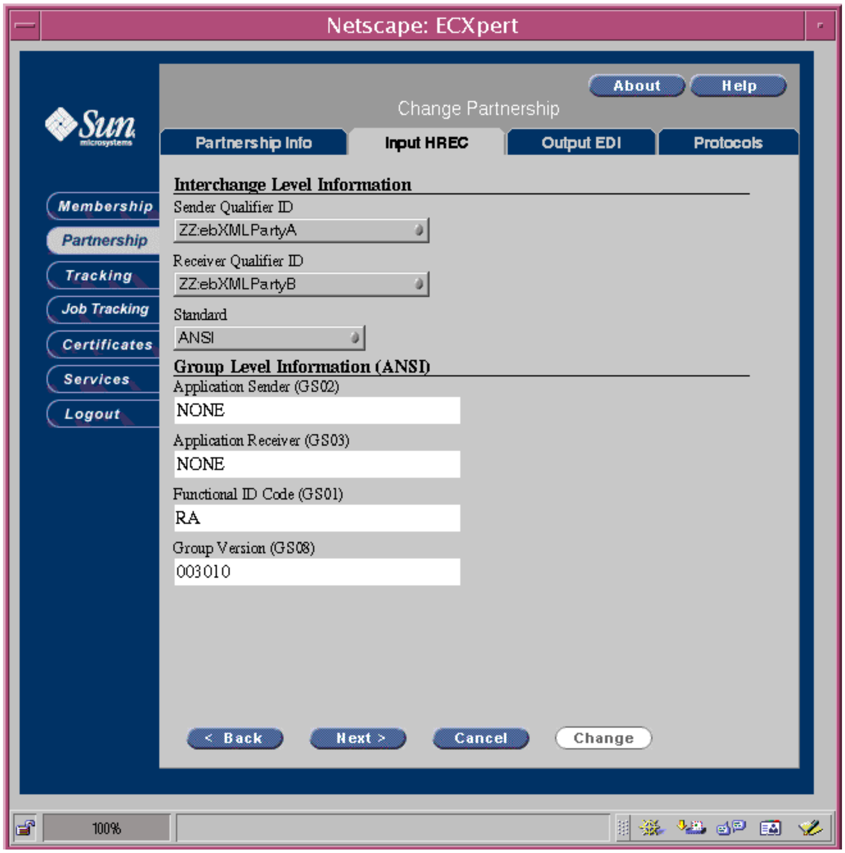
Sender Certificate Type: None

Receiver Certificate Type: None

Encryption and Authentication: Not Signed or Encrypted (plain)

3. Click Next and specify the following for Input HREC:

Field	Value
Sender Qualifier ID	`ZZ:ebXMLPartyA`
Receiver Qualifier ID	`ZZ:ebXMLPartyB`
Standard	`ANSI`
Application Sender	`NONE`
Application Receiver	`NONE`
Functional ID Code	`RA`
Group Version	`003010`



Netscape: ECXpert

Change Partnership

Interchange Level Information

Sender Qualifier ID: ZZ:ebXMLPartyA

Receiver Qualifier ID: ZZ:ebXMLPartyB

Standard: ANSI

Group Level Information (ANSI)

Application Sender (GS02): NONE

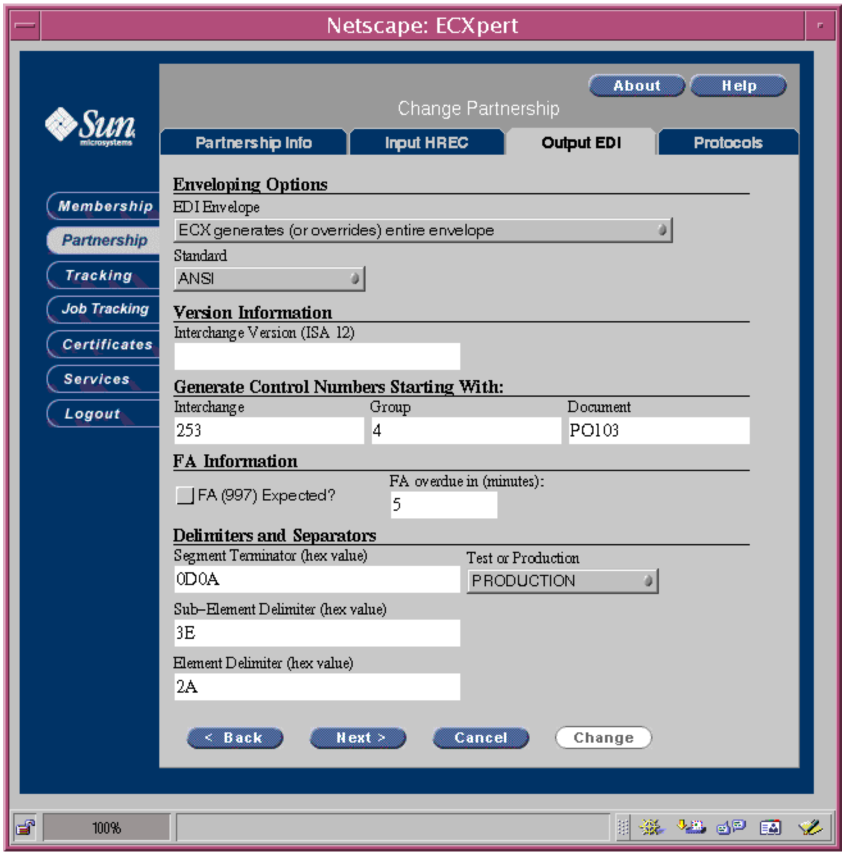
Application Receiver (GS03): NONE

Functional ID Code (GS01): RA

Group Version (GS08): 003010

4. Click Next and specify the following for Output EDI:

Field	Value
Envelope Options	
EDI Envelope	ECX generates (or overrides) entire envelope
Standard	`ANSI`
Version Information	
Interchange Version	<Leave Blank>
Generate Control Numbers Starting With	
Interchange	`253`
Group	`4`
Document	`PO103`
FA Information	
FA	<Leave unselected>
FA overdue	`5`
Delimiters and Separators	
Segment Terminator	`0D0A`
Test or Production	`PRODUCTION`
Sub-element Delimiter	`3E`
Element Delimiter	`2A`



Change Partnership

Enveloping Options

EDI Envelope: ECX generates (or overrides) entire envelope

Standard: ANSI

Version Information

Interchange Version (ISA 12):

Generate Control Numbers Starting With:

Interchange	Group	Document
253	4	PO103

FA Information

□ FA (997) Expected?

FA overdue in (minutes): 5

Delimiters and Separators

Segment Terminator (hex value): 0D0A

Test or Production: PRODUCTION

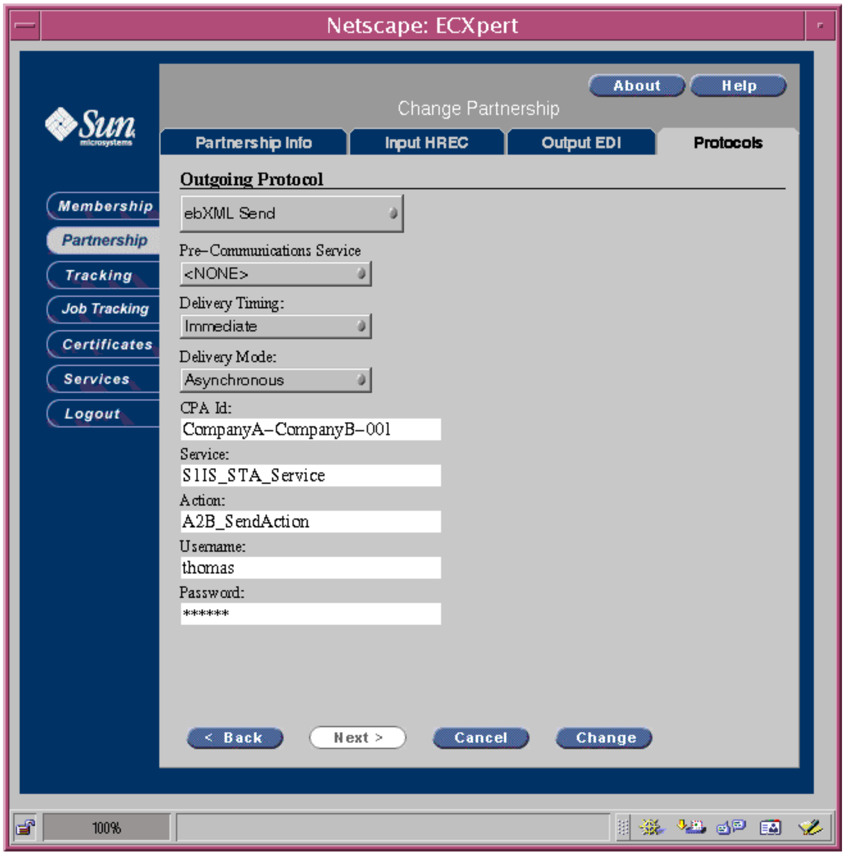
Sub-Element Delimiter (hex value): 3E

Element Delimiter (hex value): 2A

5. Select Next and specify the following for Protocols:

Field	Value
Outgoing Protocol	`ebXMLSend`
Pre-Communications Service	<NONE>
Delivery Timing	`Intermediate`
Delivery Mode	`Asynchronous`
CPA-Id	`CompanyA-CompanyB-001`

Field	Value
Service	`S1IS_STA_Service`
Action	`A2B_SendAction`
Username	*Secure Trading Agent User*
Password	*Password for Secure Trading Agent User above*



Change Partnership

Outgoing Protocol

ebXML Send

Pre-Communications Service: <NONE>

Delivery Timing: Immediate

Delivery Mode: Asynchronous

CPA Id: CompanyA-CompanyB-001

Service: S1IS_STA_Service

Action: A2B_SendAction

Username: thomas

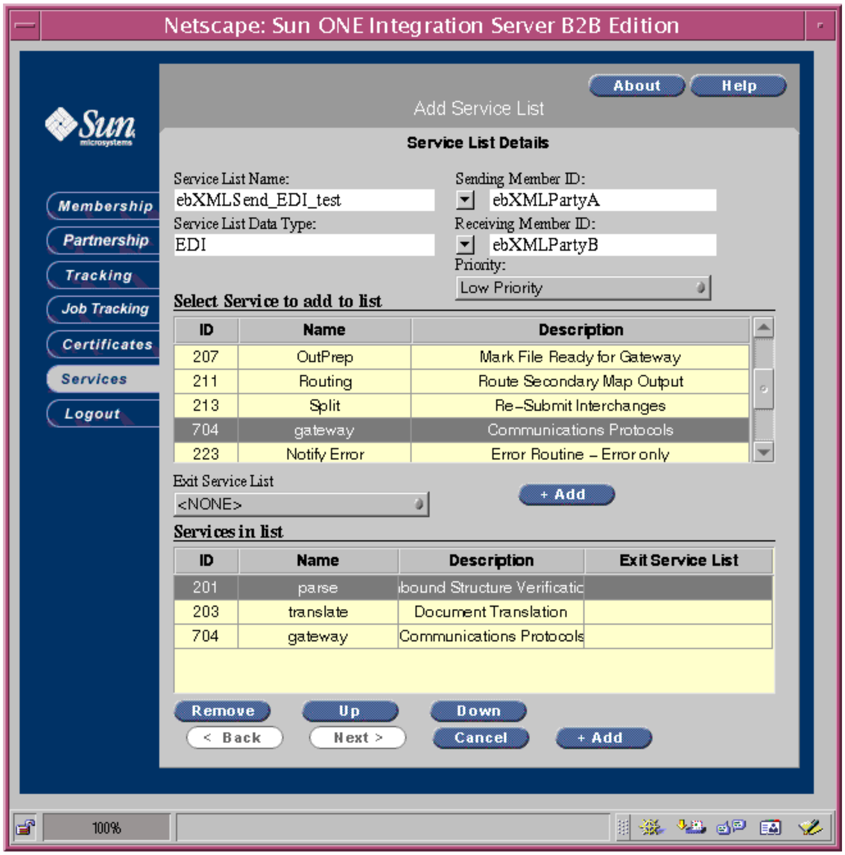
Password: ******

6. Click Change to modify the partnership with the outgoing protocol.

➤ To create a service list to route documents to the ebXML Send Agent

1. From the Support interface, select Services.
2. Select Add and specify the following:

Field	Value
Service List Details	
Service List Name	ebXMLSend_EDI_test
Service List Datatype	EDI
Sending Member ID	ebXMLPartyA
Receiving Member ID	ebXMLPartyB
Priority	Low Priority
Exit Service List	<NONE>
Services in List	parse
	translate
	gateway



Add Service List

Service List Details

Service List Name: ebXMLSend_EDI_test

Service List Data Type: EDI

Sending Member ID: ebXMLPartyA

Receiving Member ID: ebXMLPartyB

Priority: Low Priority

Select Service to add to list

ID	Name	Description
207	OutPrep	Mark File Ready for Gateway
211	Routing	Route Secondary Map Output
213	Split	Re–Submit Interchanges
704	gateway	Communications Protocols
223	Notify Error	Error Routine – Error only

Exit Service List: <NONE>

Services in list

ID	Name	Description	Exit Service List
201	parse	ıbound Structure Verificatic	
203	translate	Document Translation	
704	gateway	Communications Protocols	

3. Click Add to add the Service List.

ECXpert Set Up for Company B

This section shows how to set up a partnership for Company B to retrieve ebXML messages. It contains procedures for the following tasks:

- Creating a partnership between the submitting member and receiving member for ebXML messaging
- Setting up a service list to handle incoming ebXML documents

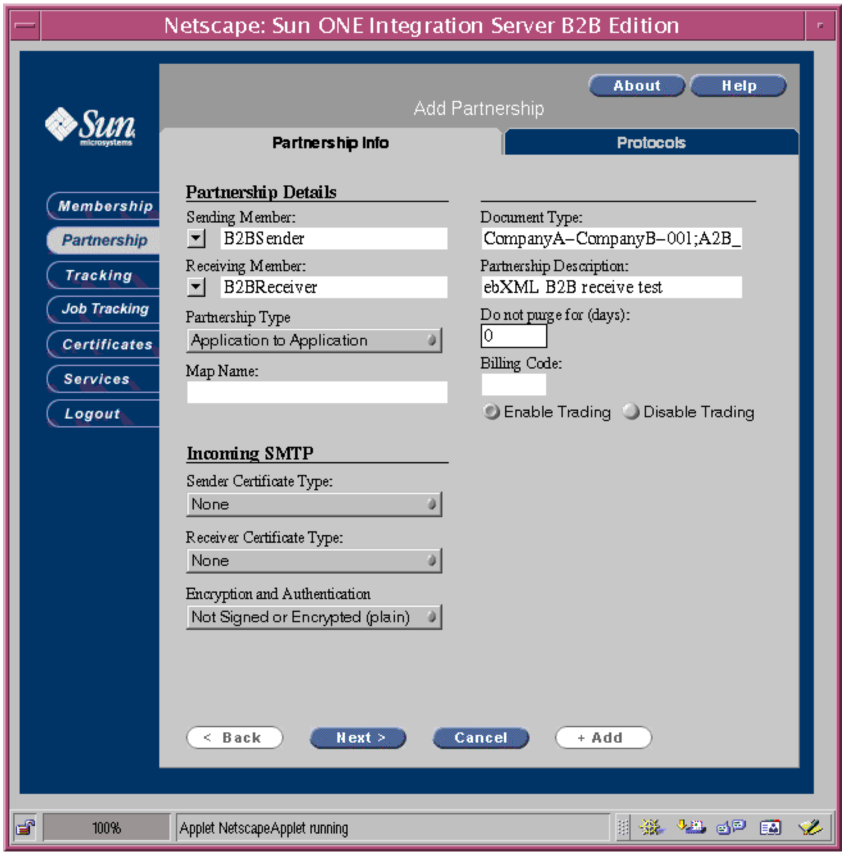
The sending member and receiving member for ebXML messaging are specified when you configure the ebXML Receive Agent. Typically, you specify these members at install time. The procedures in this section assume you specified *B2BSender* and *B2BReceiver* for these members. Modify these procedures to use the actual names you specified.

These procedures should be performed using the ECXpert installation on the machine you are using for Company B (Receiver).

➤ To create a partnership between B2BSender and B2BReceiver

1. From the Support interface, select Partnership.
2. Select Add and specify the following:

Field	Value
Sending Member	`B2BSender` (or the name used in Company B's Receive Agent configuration)
Receiving Member	`B2BReceiver` (or the name used in Company B's Receive Agent configuration)
Partnership Type	`Application to Application`
Map Name	<Leave Blank>
Document Type	`CompanyA-CompanyB-001;A2B_SendAction`
Partnership Description	`ebXML B2B receive test`
Do not purge	`0`
Billing Code	<Leave Blank>
Enable Trading	`On`
Incoming SMTP	`None, None, plain`



Add Partnership

Partnership Details

Sending Member: B2BSender

Receiving Member: B2BReceiver

Partnership Type: Application to Application

Map Name:

Document Type: CompanyA–CompanyB–001;A2B_

Partnership Description: ebXML B2B receive test

Do not purge for (days): 0

Billing Code:

○ Enable Trading ○ Disable Trading

Incoming SMTP

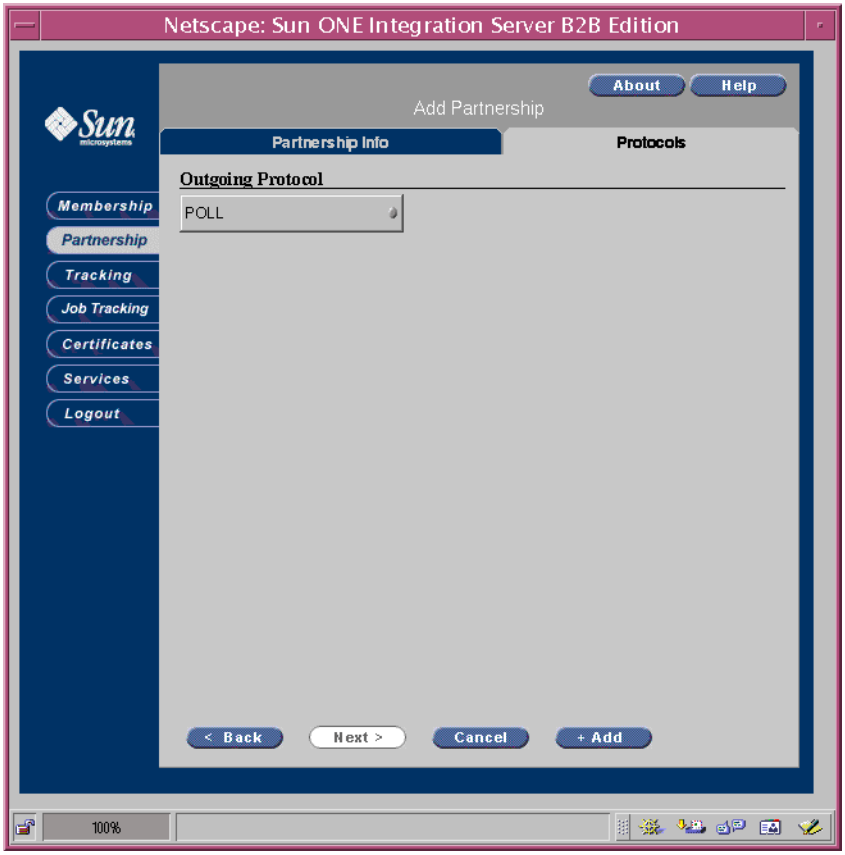
Sender Certificate Type: None

Receiver Certificate Type: None

Encryption and Authentication: Not Signed or Encrypted (plain)

Applet NetscapeApplet running

3. Select Protocols and specify `POLL` for the Outgoing Protocol.



Outgoing Protocol

POLL

4. Select Add to add the partnership.

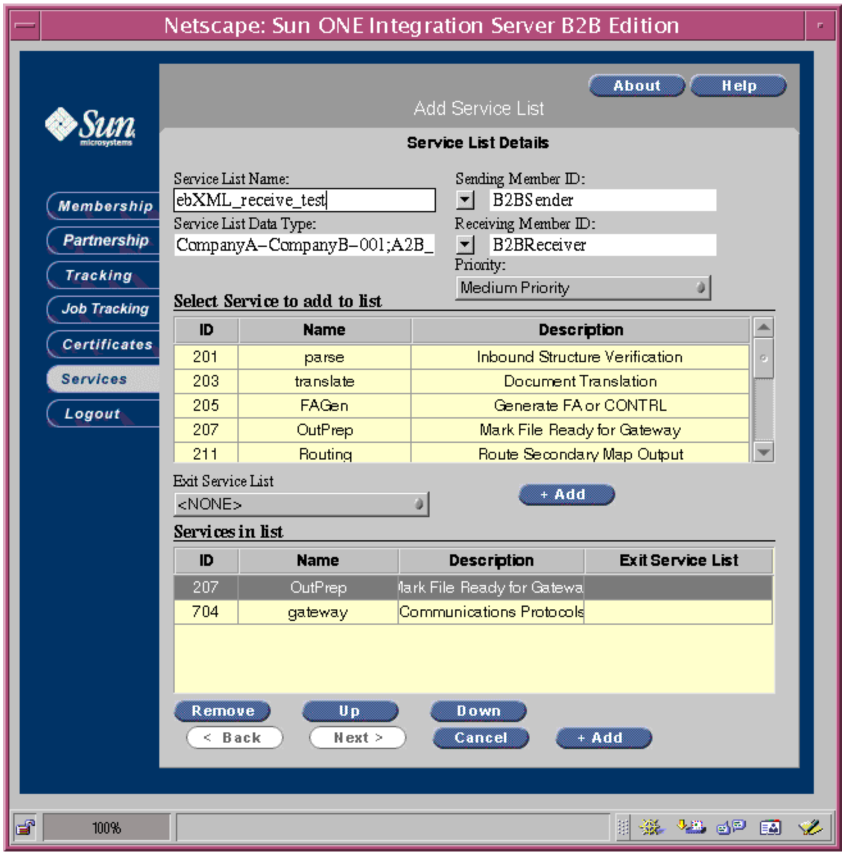
➤ To create a service list to route incoming ebXML documents

1. From the Support interface, select Services.

2. Select Add and specify the following:

Field	Value
Service List Details	
Service List Name	ebXML_receive_test
Service List Datatype	CompanyA-CompanyB-001;A2BSendAction

Field	Value
Sending Member ID	B2BSender (or the name used in your configuration)
Receiving Member ID	B2BReceiver (or the name used in your configuration)
Priority	Medium Priority
Exit Service List	<NONE>
Services in List	outprep
	gateway



Add Service List

Service List Details

Service List Name: ebXML_receive_test

Service List Data Type: CompanyA–CompanyB–001;A2B_

Sending Member ID: B2BSender

Receiving Member ID: B2BReceiver

Priority: Medium Priority

Select Service to add to list

ID	Name	Description
201	parse	Inbound Structure Verification
203	translate	Document Translation
205	FAGen	Generate FA or CONTRL
207	OutPrep	Mark File Ready for Gateway
211	Routing	Route Secondary Map Output

Exit Service List: <NONE>

Services in list

ID	Name	Description	Exit Service List
207	OutPrep	lark File Ready for Gatewa	
704	gateway	Communications Protocols	

3. Click Add to add the service

Running the ebXML Messaging Example

This section describes how you can use the ebXML messaging example to send an ebXML message from Company A to Company B. Figure 4 on page 28 shows the document flow for the example.

Before you run the example make sure you have done the following:

- Company A (Sender)
    - Installed Secure Trading Agent on the machine for Company A.
    - Installed ECXpert on the machine for Company A and configured it properly for ebXML messaging
    - Set up Secure Trading Agent for the examples, as explained in "Secure Trading Agent Set Up" on page 30.
    - Set up ECXpert for the example, as explained in "ECXpert Set Up for Company A" on page 31.
- Company B (Receiver)
    - Installed Secure Trading Agent on the machine for Company B.
    - Installed ECXpert on the machine for Company B and configured it properly for ebXML messaging
    - Set up Secure Trading Agent for the examples, as explained in "Secure Trading Agent Set Up" on page 30.
    - Set up ECXpert for the example, as explained in "ECXpert Set Up for Company B" on page 43.

The following procedure shows how Company A (Sender) submits a document to the ebXML Send Agent.

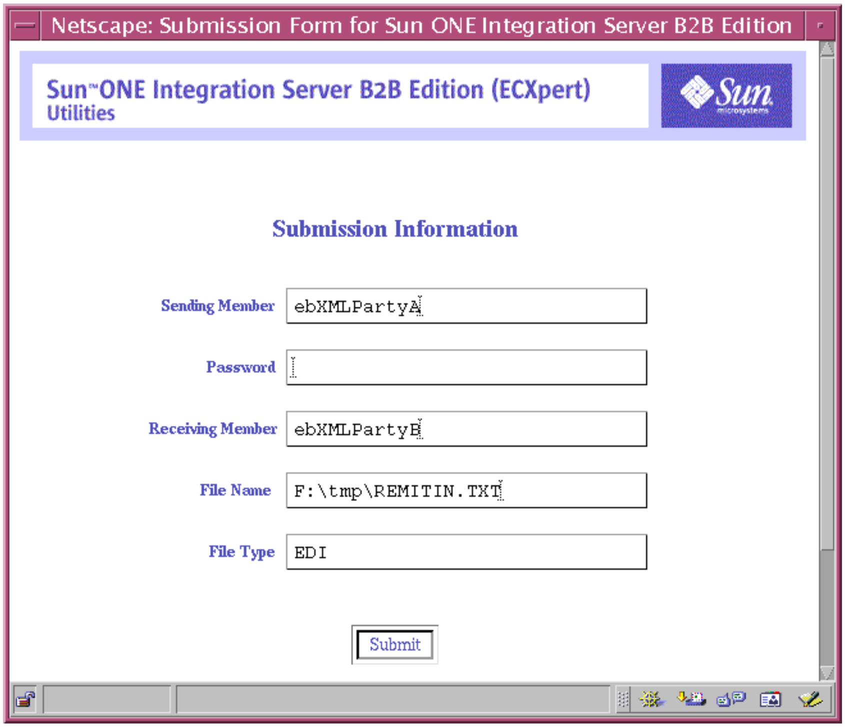
➤ To send a document from Company A as an ebXML message

1. In the ECXpert for Company A, make sure the ebXML Send Agent is On.
2. In the ECXpert for Company B, make sure the ebXML Receive Agent is On.
3. Select the Utilities Interface, and then select Document Submission Form.
4. In the Submission Form, specify the following:

Field	**Value**
Sending Member	`ebXMLPartyA`
Password:	
Receiving Member	`ebXMLPartyB`
File Name	*ExamplePath*`/REMITIN.TXT`
File Type	EDI

Specify a password only if you provided a password for the sending member when you created the member. If you have not specified a password, then the Sending Member should be trusted. *ExamplePath* is the path to a copy of document provided with this example. You should use a copy of the document because the ebXML Send Agent deletes the document after retrieving it for submission. The location of the document provided with the example is:

```
ECX_HOME/NS-apps/ECXpert/bin/ebxmltest/PartyA/REMITIN.TXT
```



Netscape: Submission Form for Sun ONE Integration Server B2B Edition

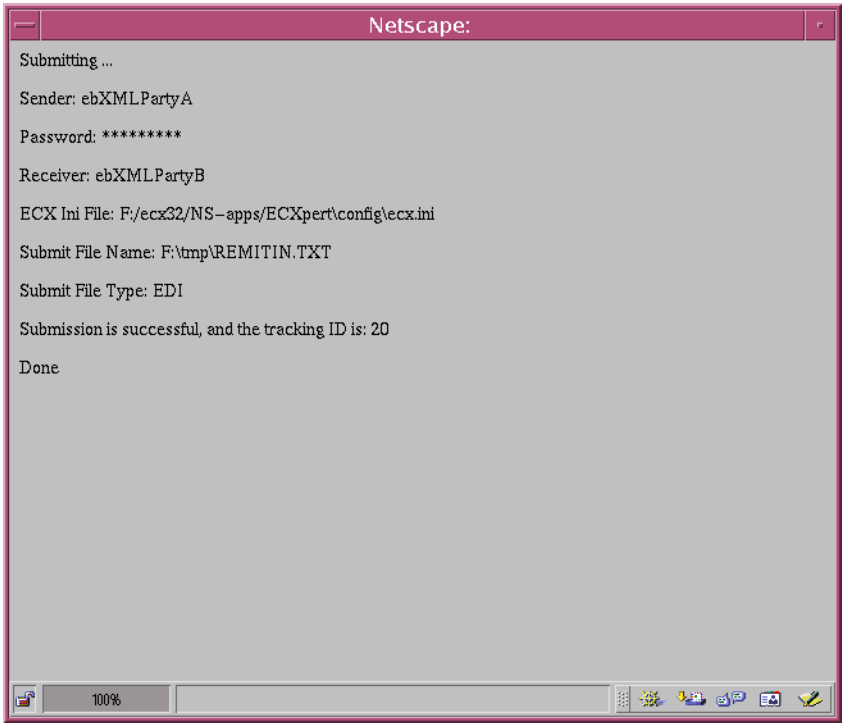
Sun™ONE Integration Server B2B Edition (ECXpert)
Utilities

Submission Information

Sending Member	ebXMLPartyA
Password	
Receiving Member	ebXMLPartyB
File Name	F:\tmp\REMITIN.TXT
File Type	EDI

5. Click `Submit` to send the document.

   A message displays indicating information about the submitted document, including the *Tracking ID*. In this example, the Tracking ID is `20`, which you specify in the next procedure to track a submitted document. Your Tracking ID may differ.



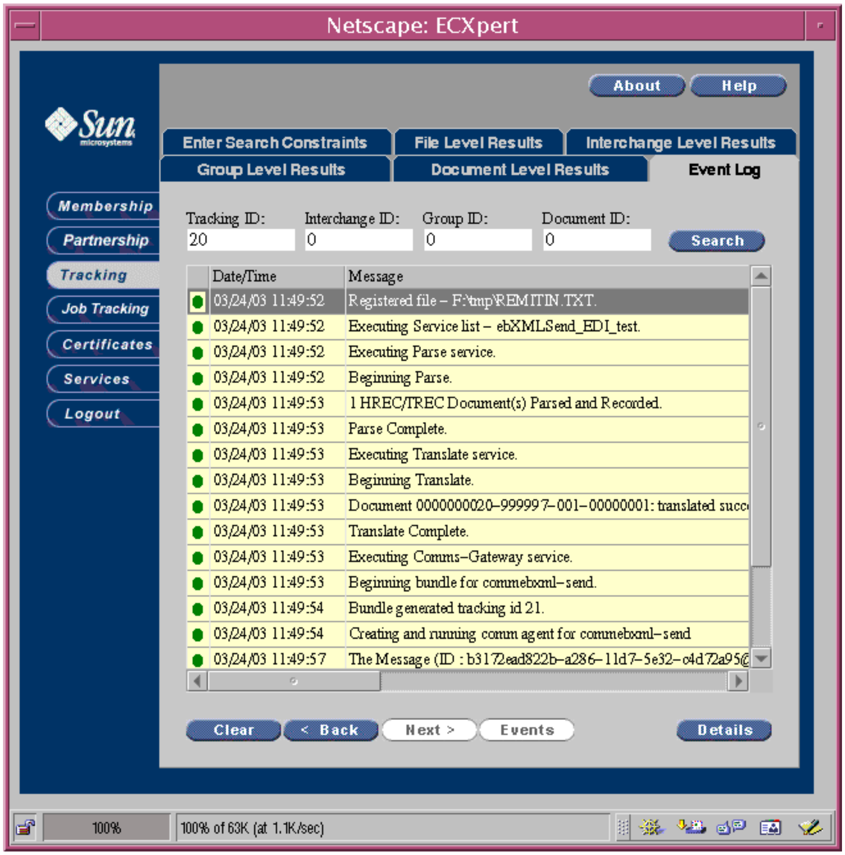
```
Submitting ...

Sender: ebXMLPartyA

Password: *********

Receiver: ebXMLPartyB

ECX Ini File: F:/ecx32/NS-apps/ECXpert\config\ecx.ini

Submit File Name: F:\tmp\REMITIN.TXT

Submit File Type: EDI

Submission is successful, and the tracking ID is: 20

Done
```

The following procedure shows how to track the document you just submitted.

➤ To track the document submitted to Company A

1. In the ECXpert for Company A, from the Support interface, select Tracking.
2. In the Event Log tab, specify the Tracking ID, as indicated below, and select Search.

   The Tracking ID was displayed when you submitted the document, as indicated in Step 5 of the previous procedure. This example uses `20` for the Tracking ID. Your Tracking ID may differ.



Tracking ID: 20
Interchange ID: 0
Group ID: 0
Document ID: 0

	Date/Time	Message
●	03/24/03 11:49:52	Registered file – F:\tmp\REMITIN.TXT.
●	03/24/03 11:49:52	Executing Service list – ebXMLSend_EDI_test.
●	03/24/03 11:49:52	Executing Parse service.
●	03/24/03 11:49:52	Beginning Parse.
●	03/24/03 11:49:53	1 HREC/TREC Document(s) Parsed and Recorded.
●	03/24/03 11:49:53	Parse Complete.
●	03/24/03 11:49:53	Executing Translate service.
●	03/24/03 11:49:53	Beginning Translate.
●	03/24/03 11:49:53	Document 0000000020–999997–001–00000001: translated succ
●	03/24/03 11:49:53	Translate Complete.
●	03/24/03 11:49:53	Executing Comms–Gateway service.
●	03/24/03 11:49:53	Beginning bundle for commebxml–send.
●	03/24/03 11:49:54	Bundle generated tracking id 21.
●	03/24/03 11:49:54	Creating and running comm agent for commebxml–send
●	03/24/03 11:49:57	The Message (ID : b3172ead822b–a286–11d7–5e32–c4d72a95@

The following procedure shows how Company B retrieves and views the document from Company B.

➤ To retrieve and view the document from Company B

1. In the ECXpert for Company B, make sure the ebXML Receive Agent is On.

2. Log in to the Support Interface.

3. Select Tracking.

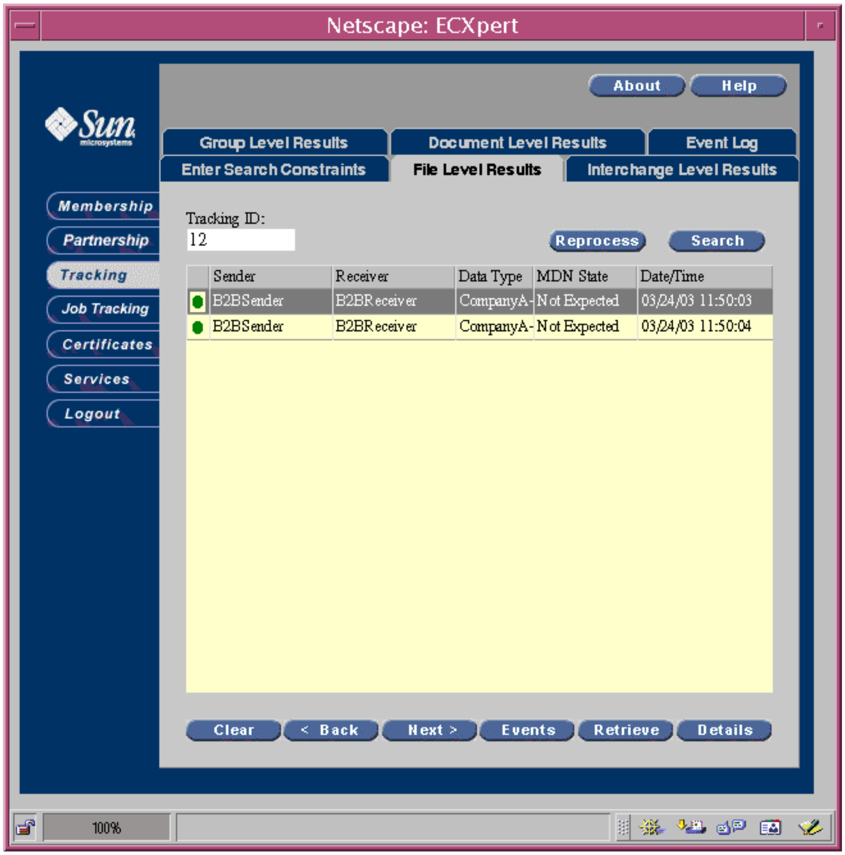
4. From the Enter Search Constraints Tab, specify the following:

Field	Value
Sending Member ID	*`B2BSender`*
Receiving Member ID	*`B2BReceiver`*
Search Level	`File`

Replace *B2BSender* and *B2BReceiver* with the actual values you specified for the B2B Sender and B2B Receiver when configuring the ebXML Receive Agent. Leave all other fields blank.

5. Select Search.

The view switches to the File Level Results tab, which lists the search results.

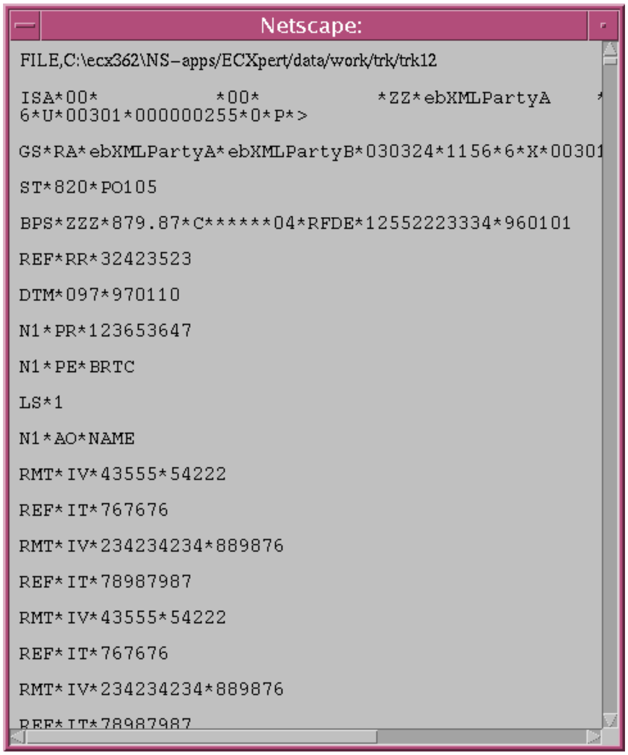


Tracking ID: 12

	Sender	Receiver	Data Type	MDN State	Date/Time
●	B2BSender	B2BReceiver	CompanyA-	Not Expected	03/24/03 11:50:03
●	B2BSender	B2BReceiver	CompanyA-	Not Expected	03/24/03 11:50:04

6. Select the first item in the list and click Retrieve.

   The retrieved document displays in a new browser window.



```
FILE,C:\ecx362\NS-apps/ECXpert/data/work/trk/trk12

ISA*00*          *00*          *ZZ*ebXMLPartyA    *
6*U*00301*000000255*0*P*>

GS*RA*ebXMLPartyA*ebXMLPartyB*030324*1156*6*X*00301

ST*820*PO105

BPS*ZZZ*879.87*C******04*RFDE*12552223334*960101

REF*RR*32423523

DTM*097*970110

N1*PR*123653647

N1*PE*BRTC

LS*1

N1*AO*NAME

RMT*IV*43555*54222

REF*IT*767676

RMT*IV*234234234*889876

REF*IT*78987987

RMT*IV*43555*54222

REF*IT*767676

RMT*IV*234234234*889876

REF*IT*78987987
```

7. For more information on the document, click Events to view the events for the specified Tracking ID.

Custom Configuration of ebXML Communications Agents

This section describes how to manually configure ebXML Communications Agents. You perform a manual configuration under the following circumstances:

- You did not specify the ebXML messaging parameters during installation of ECXpert

  This occurs if you installed ECXpert before installing Secure Trading Agent, or elected not to provide these parameters when installing ECXpert.

- You want to modify your ebXML Communications Agent configuration

  This occurs if you made an error when specifying ebXML messaging parameters, or your requirements for ebXML messaging change.

Use the System Interface to modify the ebXML Communications Agent configuration. When modifying the configuration, for various settings you must provide the path to your Secure Trading Agent installation (EBROOT). The default values for EBROOT are:

```
/opt/SUNWebXML/S1ISSTA (Solaris)
C:\Sun\S1ISSTA (Windows)
```

If you specified a custom location when installing Secure Trading Agent, make appropriate adjustments when you specify EBROOT.

The following procedures shows how to manually configure ebXML Communications Agents. If you did not specify the ebXML configuration parameters during installation, then you should perform each step in both procedures. If you are modifying an existing ebXML Communications Agent configuration, then perform only the steps necessary for your modification.

➤ To manually configure the ebXML Communications Agents

1. Open the System Interface and select Edit Section.
2. Select `commebxml-send` to configure the ebXML Send Agent, and then select Edit.
3. Modify the configuration parameters listed in Table 3 for the ebXML Send Agent.

Table 3 ebXML Send Configuration Parameters

Parameter	Description	Modification
`classpath`	Classpath	For initial configuration, replace `[EBROOT]` with the path to the Secure Trading Agent installation you are using to send ebXML messages.
`ebroot`	Path to the Secure Trading Installation	Specify EBROOT as the path to the Secure Trading Agent installation you are using to send ebXML messages.
`bootclasspath`	Classpath for the JRE. The classpath to all Secure Trading Agent Java™ Archive (JAR) files, which ECXpert appends to the bootstrap classpath.	For initial configuration, replace `[EBROOT]` with the path to the Secure Trading Agent installation you are using to send ebXML messages.

Table 3 ebXML Send Configuration Parameters *(Continued)*

Parameter	Description	Modification
`logincfg`	Path to the Secure Trading Agent configuration file	For initial configuration, replace `[EBROOT]` with the path to the Secure Trading Agent installation you are using to send ebXML messages.

4. Select the `commebxml-receive` section to modify the ebXML Receive Agent and then select Edit.

5. Modify the configuration parameters listed in Table 4 for the ebXML Receive Agent.

Table 4 ebXML Receive Configuration Parameters

Parameter	Description	Modification
`classpath`	Classpath for ECXpert and Secure Trading Agent Java™ classes.	For initial configuration, replace `[EBROOT]` with the path to the Secure Trading Agent installation you are using to receive ebXML messages.
`ebroot`	Path to the Secure Trading Installation	Specify EBROOT as the path to the Secure Trading Agent installation you are using to receive ebXML messages.
`bootclasspath`	Classpath for the JRE. The classpath to all Secure Trading Agent Java Archive (JAR) files, which ECXpert appends to the bootstrap classpath.	For initial configuration, replace `[EBROOT]` with the path to the Secure Trading Agent installation you are using to receive ebXML messages.
`logincfg`	Path to the Secure Trading Agent configuration file	For initial configuration, replace `[EBROOT]` with the path to the Secure Trading Agent installation you are using to receive ebXML messages.
`ebxml_user`	System user that logs into Secure Trading Agent	Provide a valid user that can log into Secure Trading Agent. This user does not have to be a Secure Trading Agent administrator.
`ebxml_passwd`	Password for the system user that logs into Secure Trading Agent	Encrypted password for the `ebxml_user`. Use the password utility, `bdgsetpasswd`, to encrypt the password and update the configuration file. For more information, refer to "Password Utility" on page 56.

Table 4 ebXML Receive Configuration Parameters *(Continued)*

Parameter	Description	Modification
`ebxml_b2b_user`	The *ebXML B2B User*. The ebXML Receive Agent uses this parameter to retrieve messages from Secure Trading Agent.	The ebXML B2B User corresponds to a B2B User defined in Secure Trading Agent. The B2B User in Secure Trading is associated with agreement roles in deployed ebXML agreements.
`b2b_sender`	Sending member for submitting payloads	Sender for an ECXpert partnership used for ebXML messaging. If you did not specify a value for `b2b_sender` at install time, you must create this member before this value takes effect. If you have Parse in your service list, and the sender specified in the payload does not match the sender used for submission, then the `b2b_sender` must be an ECXpert administrator.
`b2b_sender_passwd`	Password for the `b2b_sender`	Encrypted password for the `b2b_sender`. Use the password utility, `bdgsetpasswd`, to encrypt the password and update the configuration file. For more information, refer to "Password Utility" on page 56. If you do not specify a password, then the sender should be trusted.
`b2b_receiver`	Receiving member for submitting ebXML payloads	Receiver for an ECXpert partnership that is used for ebXML messaging. If you did not specify a value for `b2b_receiver` at install time, you must create this member before this value takes effect.

`filetype_format`	The Receive Communications Agent uses the File Type Format to construct the filetype submission parameter when submitting retrieved messages.	Build a File Type Format from the following values:
		`cpa-id`
		`service-name`
		`action-name`
		The File Type Format must contain at least one value. Use a semi-colon as a delimiter between values. The recommended value for filetype_format is:
		`cpa-id;action-name`

6. After specifying configuration values, restart the ebXML Communications Agents for the new settings to take effect.

NOTE You can edit the ecx.ini file directly to modify your configuration. The ecx.ini file has comments to guide you through modification of this file.

Password Utility

ECXpert provides the bdgsetpassword utility to encrypt passwords specified in the ECXpert configuration file (ecx.ini). This utility is necessary to configure the password for Secure Trading User (ebxml_passwd) and the password for the sending member for ebXML messaging (b2b_sender).

The bdgsetpassword utility can be found at the following location:

```
ECX_HOME/NS-apps/ECXpert/bin
```

ECX_HOME is the directory in which you installed ECXpert.

Environment Variables

Before you run the utility, modify your environment as follows:

On Solaris and Windows platforms, set an environment variable, BDGHOME, to point to:

```
ECX_HOME/NS-apps/ECXpert
```

On Solaris platforms, modify LD_LIBRARY_PATH to include:

```
ECX_HOME/NS-apps/ECXpert/lib
```

On Windows platforms, modify PATH to include:

```
ECX_HOME/NS-apps/ECXpert/lib
```

ECX_HOME is the directory in which you installed ECXpert.

Usage

When you run the bdgsetpasswd utility, you must specify which password you are generating, as indicated below:

```
bdgsetpasswd -i path/ecx.ini -ebp password
bdgsetpasswd -i path/ecx.ini -b2bp password
```

-ebp: generate Secure Trading User password (`ebxml_passwd`)
-b2bp: generate ebXML sending member password (`b2b_sender`)
path: `ECX_HOME/NS-apps/ECXpert/config`
password: the password you want to encrypt for the configuration file

The utility automatically updates the `ecx.ini` file. After running this utility, you must restart the ebXML Receive Agent for the changes to take effect.

Troubleshooting

You can use the B2B Event Log to troubleshoot errors encountered when sending or receiving messages.

➤ **To view the B2B Event Log**

1. From the Support Interface, select Tracking.
2. Select the Event Log Tab.
3. To view Receive error events, specify `0` (zero) for the tracking ID.
4. To view Send error events, specify a specific tracking ID.

For more information on viewing the B2B Event Log, refer to the Chapter "Tracking the Documents that ECXpert Processes," available from the *ECXpert Administrator's Guide*.

Table 5 lists ebXML error event numbers, the message displayed, and possible causes and remedies.

Table 5 ebXML Messaging Error Events

Error	Description
19001	The required parameter is missing in the configuration file. **Possible Causes:** The ebXML Send Communications Agent section in the `ecx.ini` file is missing a required parameter. **Remedies:** Verify that all required parameters are specified in the `commebxml-send` section of the `ecx.ini` file. Refer to "Custom Configuration of ebXML Communications Agents" on page 52 for information on modifying your configuration.

Table 5 ebXML Messaging Error Events *(Continued)*

Error	Description
19002	The ebXML Login Configuration file does not exist. **Possible Causes:** The ebXML Send Communications Agent section in the `ecx.ini` has an invalid value for the `logincfg` parameter. **Remedies:** Specify a valid path for this parameter.
19003	An unexpected error occurred. The Message could not be sent. **Possible Causes:** An error occurred when sending processed output file using Secure Trading Agent. This failure can indicate: • Secure Trading Agent configuration problems, such as CPA not deployed, Secure Trading Agent database connection broken, or other error in Secure Trading Agent. • The ebXML Send Communications Agent is configured with improper parameters on the Partnership Protocols tab. **Remedies:** Refer to the Secure Trading Agent documentation for information on correctly configuring, using, and troubleshooting Secure Trading Agent. Refer to "Sending ebXML Messages" on page 23 for information on specifying partnership protocol information.
19004	The processed output file does not exist. **Possible Causes:** An error occurred in the execution of the service list and there is no output file to be sent or the processed output file was accidentally deleted. **Remedies:** Fix the error that caused the service list execution to fail.
19005	The EBROOT location does not exist. **Possible Causes:** The value of the ebroot parameter in the `ecx.ini` is incorrect. **Remedies:** Modify `ecx.ini` with the correct path to the Secure Trading Installation. Refer to "Custom Configuration of ebXML Communications Agents" on page 52 for information on modifying your configuration.
19006	An error occurred in creating the JVM. **Possible Causes:** An internal error that occurs when the ebXML Send Agent cannot create a Java Virtual Machine. **Remedies:** Verify the classpath parameters in the `ecx.ini` file.
19007	An error occurred in attaching to the JVM. **Possible Causes:** An internal error that occurs when the ebXML Send Agent thread cannot attach to a Java Virtual Machine. **Remedies:** Contact Support. This could indicate resource issues.

Table 5 ebXML Messaging Error Events *(Continued)*

Error	Description
19008	Mandatory ebXML parameter missing. **Possible Causes:** An ebXML parameter is missing from the Partnership Protocols tab. **Remedies:** Specify parameters correctly on the Partnership Protocols tab, as discussed in "Sending ebXML Messages" on page 23.
19009	ebXML Receive component missing configuration parameters. **Possible Causes:** An ebXML parameter is missing from the ebXML Receive Communications Agent section in the `ecx.ini` file. **Remedies:** Verify that all required parameters are specified in the `commebxml-receive` section of the `ecx.ini` file.