SeeBeyond ICAN Suite

# DialUp e*Way Intelligent Adapter User's Guide

*Release 5.0.5 for Schema Run-time Environment (SRE)*

SEEBEYOND®

Version 20050405233038.

# Contents

Chapter 1

# Introduction

This chapter includes a brief description of SeeBeyond™ Technology Corporation's (SeeBeyond™) Dial-up e*Way Intelligent Adapter (Dial-up e*Way), as well as system requirements for using the e*Way.

## 1.1 Intended Reader

The reader of this guide is presumed to be a developer or System Administrator with responsibility for maintaining the e*Gate system; to have expert-level knowledge of Windows operations and administration; to be thoroughly familiar with the use of e*Gate Integrator, having a good knowledge of how to use the Java Collaboration editor and Windows-style GUI operations.

## 1.2 Overview

The Dial-up e*Way provides the user of e*Gate Integrator to dial out to a remote server and perform the following functionality:

- Support a variety of protocols, such as Kermit, x-modem, y-modem, and z-modem
- Support parameter based configuration of a connection
- Support identification of duplicate upload/download of files
- Support multiple upload/download over a single connection
- Support for monitoring the results of file upload/download, connection attempts, such as session logging

## 1.3 Functional Description

The Dial-up e*Way dials out using the local modem to a remote machine, logs into the remote machine, and then performs the necessary file upload/download operations. The e*Way allows the user to use the e*Way connection configuration to configure the modem, the login mode and the various file transfer options.

### 1.3.1 Components

The following components comprise the Java-based Dial-up e*Way

#### Dialup.jar

Contains the logic required by the e*Way to parse the input Events and perform the required dialing and file transfer operations.

#### Dialup.xsc

Allows the user to create the Event Type Definition within the Collaboration editor, access the methods defined within it in conjunction with the parsing engine contained within the extended Java Collaboration Service.

#### e*Way Connection

The Dial-up e*Way connection provides access to the information necessary for configuring the modem and performing the required dial-up operations.

## 1.4 Supported Operating Systems

The Dial-up e*Way is supported on the following platforms:

- Windows 2000, Windows XP, and Windows Server 2003
- HP Tru64 V5.1A
- IBM AIX 5.1L and 5.2
- Sun Solaris 8 and 9

## 1.5 System Requirements

To use the Dial-up e*Way, you need the following system requirements:

- An e*Gate Participating Host.
- The amounts of disk space required on both the Participating and the Registry Host vary. See the **Readme.txt** file in the root directory of the e*Gate installation CD-ROM, for specific version information.

***Note:*** *Additional disk space is required to process and queue the data that this e*Way processes; the amount necessary varies based on the type and size of the data being processed, and any external applications performing the processing.*

- A TCP/IP network connection, a phone line, and so on.

## 1.6 External System Requirements

The Dial-up e*Way can be used on Windows in "client" mode only, that is, to dial out from the Windows machine. The Dial-up e*Way can dial into, or out from UNIX machines.

To enable the e*Way to communicate correctly with the remote server, you need the following external requirements:

- Modem connection.
- The necessary Kermit, X, Y, or Z software to support file transfers.

### 1.6.1 Logging

In general, an e*Way relays any pertinent information as to the state, protocol position, and any conditions that are helpful to the user to understand what is taking place according to the debug level settings, and either logs the information to a file, or notifies the Alert Agent.

Standard debug levels are set through the standard user interface.

### 1.6.2 Errors

Any error condition are written to the log file. The inability to write to the log file or any fatal/unrecoverable errors result in the e*Way shutting down after it sends an alert to the Alert Agent.

### 1.6.3 Alerting

Any errors that affect the operation of the e*Way preventing the successful delivery of a message cause an alert to be sent. If the alert can not be sent, the e*Way shuts down.

Chapter 2

# Installation

This chapter explains procedures for installing the Dial-up e*Way.

## 2.1 Installation on Windows Systems

### 2.1.1 Pre-installation

- Exit all Windows programs before running the setup program, including any anti-virus applications.
- You must have Administrator privileges to install this e*Way.

### 2.1.2 Installation Procedure

**To install the Dial-up e*Way on Windows systems**

1. Log in as an Administrator to the workstation on which you are installing the e*Way.
2. Insert the e*Way installation CD-ROM into the CD-ROM drive.
3. If the CD-ROM drive's Autorun feature is enabled, the setup application launches automatically; skip ahead to step 4. Otherwise, use the Windows Explorer or the Control Panel's **Add/Remove Applications** feature to launch the file **setup.exe** on the CD-ROM drive.
4. The InstallShield setup application launches. Follow the installation instructions until you come to the **Please choose the product to install** dialog box.
5. Select **e*Gate Integrator**, then click **Next**.
6. Follow the on-screen instructions until you come to the second **Please choose the product to install** dialog box.
7. Clear the check boxes for all selections except **Add-ons**, and then click **Next**.

8. Follow the on-screen instructions until you come to the **Select Components** dialog box.
9. Highlight (but do not check) **e*Ways**, and then click the **Change** button. The **SelectSub-components** dialog box appears.
10. Select the **Dial-up e*Way**. Click the continue button to return to the **Select Components** dialog box, then click **Next**.
11. Follow the rest of the on-screen instructions to install the Dial-up e*Way. Be sure to install the e*Way files in the suggested client installation directory. The installation utility detects and suggests the appropriate installation directory. **Unless you are directed to do so by SeeBeyond support personnel, do not change the suggested installation directory setting.**

***Note:*** *Once you have installed and configured this e*Way, you must incorporate it into a schema by defining and associating the appropriate Collaborations, Collaboration Rules, IQs, and Event Types before this e*Way can perform its intended functions. For more information about any of these procedures, please see the online Help.*

*For more information about configuring e*Ways or how to use the e*Way Editor, see the **e*Gate Integrator User's Guide**.*

## 2.2 Installation on UNIX Systems

### 2.2.1 Pre-installation

You do not require root privileges to install this e*Way. Log in under the user name that you wish to own the e*Way files. Be sure that this user has sufficient privileges to create files in the e*Gate directory tree.

### 2.2.2 Installation Procedure

**To install the Dial-up e*Way on a UNIX system**

1. Log in on the workstation containing the CD-ROM drive, and insert the CD-ROM into the drive.
2. If necessary, mount the CD-ROM drive.
3. At the shell prompt, type

   **cd /cdrom**
4. Start the installation script by typing

   **setup.sh**
5. A menu of options will appear. Select the **Install e*Way** option. Then, follow the additional on-screen directions.

**Note:** *Be sure to install the e*Way files in the suggested* ***client*** *installation directory. The installation utility detects and suggests the appropriate installation directory.* ***Unless you are directed to do so by SeeBeyond support personnel, do not change the suggested "installation directory" setting.***

6 After installation is complete, exit the installation utility and launch the Schema Designer.

**Note:** *Once you have installed and configured this e*Way, you must incorporate it into a schema by defining and associating the appropriate Collaborations, Collaboration Rules, IQs, and Event Types before this e*Way can perform its intended functions. For more information about any of these procedures, please see the online Help system.*

*For more information about configuring e*Ways or how to use the e*Way Editor, see the* ***e*Gate Integrator User's Guide****.*

## 2.3 Files/Directories Created by Installation

The Dial-up e*Way installation process installs the files shown in Table 1, within the e*Gate directory tree. Files are installed within the **eGate\server** tree on the Participating Host and committed to the "default" schema on the Registry Host.

**Table 1** Files Created by Installation

| Directories | Files |
|---|---|
| etd\dialup | Dialup.jar<br>Dialup.xsc |
| etd\ | dialupew.ctl |
| ThirdParty\antlr-2.7.1\ | Antlr.jar<br>Antlrall.jar |
| ThirdParty\Serialio\ | Serialio.jar<br>SerialPortLocal.jar<br>Kermit.jar<br>Xmodem.jar<br>Zmodem.jar |
| ThirdParty\xml\apache\ | Xerces.jar |
| ThirdParty\xml14j\ | jdom.jar |
| configs\dialup\ | Dialup.def |

The **.dll** files and shared libraries shown in Table 2 are specific to the operating system and are installed on the client machine during the installation.

**Table 2** Operating System-specific Installation Files

| Operating Systems | Directories | Files |
|---|---|---|
| Win 2000 | bin\win32 | JspWin.dll |
| HP/UX 11 | bin\hpux11 | LibjspHpxPaRisc.sl |
| Sun Solaris | bin\sparc26 | LibjspSolSparc.so |
| Compaq Tru64 | bin\ctru64_4 | LibjspTru64Alpha.so |
| AIX | bin\aix43 | libjspAixPpc.so |

Chapter 3

# e*Way Connection Configuration

This chapter describes how to configure the Java-enabled Dial-up e*Way Connection Configuration.

## 3.1 Configuring e*Way Connections

e*Way Connections are set using the Schema Designer.

**To create and configure e*Way Connections:**

1. In the Schema Designer's **Component** editor, select the **e*Way Connections** folder.
2. On the palette, click the **Create a New e*Way Connection** button.
3. The **New e*Way Connection Component** dialog box opens, enter a name for the new **e*Way Connection**. Click **OK**.
4. Double-click on the new **e*Way Connection**. For this example, the connection has been defined as **dialUpConn**.
5. The **e*Way Connection Properties** dialog box opens.
6. From the **e*Way Connection Type** drop-down box, select **Dialup**.
7. Enter the **Event Type "get"** interval in the dialog box provided. The configured default is -1 milliseconds.
8. From the **e*Way Connection Configuration File**, click **New** to create a new Configuration File for this e*Way Connection. (To use an existing file, click **Find**.)
9. The **e*Way Connection Edit Settings** window opens. Make any necessary changes to the Dial-up e*Way Connection parameters.
10. Go to **File**, **Save** to save settings.
11. Go to **File**, **Promote to Run Time**.

The Dial-up e*Way Connection configuration parameters are organized into the following sections:

- Connector
- SerialPort Settings
- Modem Settings
- Autoconnect
- ProtocolPrompts

## 3.1.1 Connector

This section contains a set of top level parameters:

- type
- class

### Type

**Description**

Specifies the type of connector.

**Required Values**

A string . The value always defaults to **Dialup** for Dial-up connections.

### Class

**Description**

Specifies the class name of the Dial-up connector object used by the e*Way.

**Required Values**

A valid package name. The default is com.stc.eways.dialup.DialupConnector.

## 3.1.2 Serial Port

This section contains the following top level parameters:

- Device name

### Device Name

**Description**

Specifies the Device Name of the serial port used by the modem on the client machine.

#### Required Values

**COM2** or **COM3**. The default is **COM3**. Can be any of the COM ports, which are available on the client machine. Check the client hardware configuration to determine on which port the modem is connected to and use that port.

### 3.1.3 Modem

This section contains the following top level parameters:

- BitRate
- DataBits
- StopBits
- Parity
- TransferMode
- OverwriteExitingFile
- Protocol
- LoginPrompt
- PasswordPrompt
- CommandPrompt

### BitRate

#### Description

Specifies the speeds with which the transmitter and the receiving modem communicate with each other.

#### Required Values

**9600**, **19200**, **38400**, **57600**, or **115200**. Select one of the available speeds.

### DataBit

#### Description

Specifies the Bit Rate at which the transmitter sends the data to the receiver. Both the transmitter and receiver must agree on the number of data bits, as well as the baud rate.

#### Required Values

**5**, **6**, **7**, or **8**. The default is **8**. Select one of the available data bit rates. Most devices transmit data using either 7 or 8 data bits.

## StopBits

### Description

Specifies the Stop Bits sent after the data bits are transmitted.

### Required Values

**1** or **2**. Select one of the available stop bits with which the transmitting and the receiving modem communicate.

## Parity

### Description

Specifies an additional bit called a parity bit, which is transmitted along with the data, besides the synchronization provided by the use of start and stop bits. It adds a small amount of error checking, to help detect data corruption that might occur during transmission.

### Required Values

ODD, MARK, SPACE, or NONE. Select one of the available choices.

## TransferMode

### Description

Specifies the file content mode to use for file transfer.

### Required Values

**TEXT** or **BINARY**. Select one of the available choices.

## OverwriteExistingFile

### Description

Specifies a flag to indicate whether to overwrite existing file during the file transfer.

### Required Values

Yes or No. Select one of the available choices. If this is set to Yes, the e*Way overwrites existing files with the same name during the upload or download. If this is set to No, during the file transfer, the e*Way checks whether a file with the same name exists. If Yes, it will skip the file transfer and go to the next operation in sequence.

## Protocol

### Description

Specifies the communication protocol to use for the file transfer.

#### Required Values

kermit , xmodem, ymodem, or zmodem. Select one of the available choices. The default is Kermit.

### LoginPrompt

#### Description

Specifies the login prompt that the Dial-up e*Way expects at the remote machine.

#### Required Values

**login** or the appropriate value**.** Enter the value of the login prompt which the e*Way expects it needs to login to the remote machine before it proceeds with the file transfer.

### PasswordPrompt

#### Description

Specifies the password prompt that the Dial-up e*Way expects at the remote machine.

#### Required Values

**Password** or the appropriate value**.** Enter the value of the password prompt which the e*Way expects it needs to login to the remote machine before it proceeds with the file transfer.

### CommandPrompt

#### Description

Specifies the command prompt that the Dial-up e*Way expects at the remote machine one it successfully logs in.

#### Required Values

The appropriate command prompt**.** Enter the value of the command prompt which the e*Way expects it needs to enter commands to the remote machine before it proceeds with the file transfer.

## 3.1.4 AutoConnectFlag Settings

The Dial-up e*Way can operate in an Auto Connect mode which requires the phone number, userid, password and machine name parameters from the configuration file. In the Auto Connect mode, this flag is set to Yes, values must then be specified in the dialog boxes below the Auto Connect Flag. If the flag is set to No, the user must provide the above parameters in the incoming Event or specify them in the Collaboration Rule.

In the Auto Connect Mode, (flag set to Yes), the e*Way will not disconnect the phone connection unless it is explicitly stopped by the user. Thus this mode is usually used when the user wants to perform multiple transfers to the same connection/number.

When the Auto Connect is set to No, the user must specify the connection parameters in the input Event. In this case, the node to disconnect the modem must also be specified. This mode is generally used to connect to different numbers and perform file transfers with the same input Event.

This section contains the following top level parameters:

- AutoConnectFlag
- PhoneNumber
- UserId
- Password
- Machine

## AutoConnectFlag

### Description

Specifies where to pick up the modem connect parameters..

### Required Values

**Yes** or **No.** The default is No.

## PhoneNumber

### Description

Specifies the phone number to dial for the file transfer.

### Required Values

A valid exiting phone number. or the appropriate value**.**

## UserId

### Description

Specifies the user ID with which the e*Way logs into the remote machine.

### Required Values

A valid user ID.

## Password

### Description

Specifies the password for the above mentioned user ID.

### Required Values

The correct password for the above mentioned user ID.

### Machine

#### Description

Specifies the machine name where the e*Way connects.

#### Required Values

A valid machine name.

## 3.1.5 ProtocolPrompts

This section contains the following top level parameters:

- XReceive
- XSend
- YReceive
- YSend
- ZReceive
- ZSend

### XReceive

#### Description

Specifies the prompt which appears after X modem receive is started.

#### Required Values

**Ready to receive** or any appropriate message.

### XSend

#### Description

Specifies prompt which appears after X modem send starts.

#### Required Values

**Start your local XMODEM receive** or **Give your local XMODEM receive command now**, or another appropriate value**.**

### YReceive

#### Description

Specifies the prompt which appears after Y modem receive starts.

#### Required Values

**rb ready**, **waiting to receive** or an appropriate value**.**

## YSend

### Description

Specifies the prompt which appears after the Y modem send starts.

### Required Values

**Start your local YMODEM receive** or an appropriate value**.**

## ZReceive

### Description

Specifies the prompt which appears after Zmodem receive starts.

### Required Values

**rz ready**, **waiting to receive** or an appropriate value**.**

## ZSend

### Description

Specifies the prompt which appears after the Z modem send starts.

### Required Values

An appropriate value**.**

Chapter 4

# Implementation

This chapter includes information pertinent to implementing the Dial-up e*Way in a production environment. Also included is a sample schema.

The following assumptions are applicable to this implementation: 1) The e*Way has been successfully installed. and 2) All necessary .jar files are accessible.

## 4.1 Dial-up Client Sample Implementation Overview

During installation, the host and Control Broker are automatically created and configured. The default name of each is the name of the host on which you are installing the e*Gate Schema Designer GUI.

### 4.1.1 The Dial-up Event Type Definition

To implement the Dial-Up e*Way the user must create an input Event Type Definition and map it to the corresponding fields to those of the Dial-Up event type definition.

The input Event Type Definition can be created manually or by using the existing DTD wizard.

The user then has to map the nodes of his input Event Type Definition to the dialup Event Type Definition (dialup.xsc) in the Java collaboration editor.

When the user is creating his collaboration rules using the dialup.xsc, he must invoke the "process()" method of the XSC in order to start the file transfer process, this method is explained in greater detail in **"Methods in the XSC file" on page 24**.

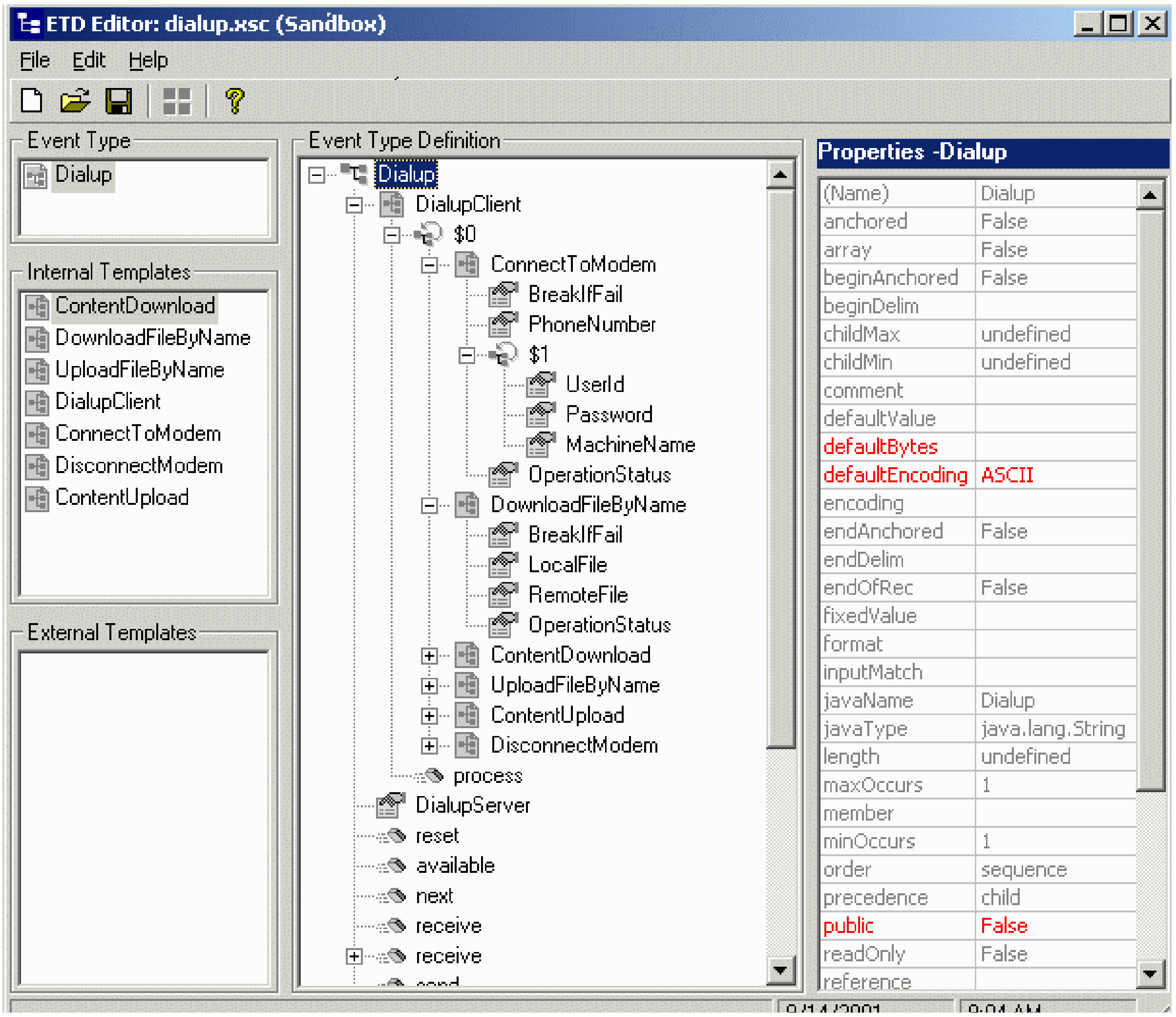
The dialup Event Type Definition has been generated from a DTD using the DTD Wizard, which comes with the XML toolkit addon in the e*Gate Addon Installation. This same DTD can also be used to create an input Event Type Definition, which can then be used to map the input Event nodes to the nodes of the dialup Event Type Definition.

The structure of the Input DTD is as shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by
cengland@seebeyond.com (SeeBeyond) -->
<!ELEMENT Dialup (DialupClient?, DialupServer?)>
```

```
<!ELEMENT DialupClient ((ConnectToModem | DownloadFileByName |
ContentDownload | UploadFileByName | ContentUpload |
DisconnectModem)+)>
<!ELEMENT ConnectToModem (PhoneNumber, (UserId | Password |
MachineName)*, OperationStatus?)>
<!ELEMENT PhoneNumber (#PCDATA)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT Password (#PCDATA)>
<!ATTLIST ConnectToModem
    BreakIfFail (yes | no) "yes"
>
<!ELEMENT MachineName (#PCDATA)>
<!ELEMENT UploadFileByName (LocalFile, RemoteFile, OperationStatus?)>
<!ATTLIST UploadFileByName
    BreakIfFail (yes | no) "yes"
>
<!ELEMENT ContentUpload (Data, RemoteFile, OperationStatus?)>
<!ATTLIST ContentUpload
    BreakIfFail (yes | no) "yes"
>
<!ELEMENT DownloadFileByName (LocalFile, RemoteFile,
OperationStatus?)>
<!ATTLIST DownloadFileByName
    BreakIfFail (yes | no) "yes"
>
<!ELEMENT ContentDownload (Data, RemoteFile, OperationStatus?)>
<!ATTLIST ContentDownload
    BreakIfFail (yes | no) "yes"
>
<!ELEMENT Data (#PCDATA)>
<!ELEMENT LocalFile (#PCDATA)>
<!ELEMENT RemoteFile (#PCDATA)>
<!ELEMENT OperationStatus (#PCDATA)>
<!ELEMENT DisconnectModem (OperationStatus)>
<!ATTLIST DisconnectModem
    BreakIfFail (yes | no) "yes"
>
<!ELEMENT DialupServer (#PCDATA)>
```

**Figure 1** Dial-up Event Type Definition Format



Event Type:

- Dialup

Internal Templates:

- ContentDownload
- DownloadFileByName
- UploadFileByName
- DialupClient
- ConnectToModem
- DisconnectModem
- ContentUpload

```
Dialup
  DialupClient
    $0
      ConnectToModem
        BreakIfFail
        PhoneNumber
        $1
          UserId
          Password
          MachineName
        OperationStatus
      DownloadFileByName
        BreakIfFail
        LocalFile
        RemoteFile
        OperationStatus
      ContentDownload
      UploadFileByName
      ContentUpload
      DisconnectModem
    process
  DialupServer
  reset
  available
  next
  receive
  receive
```

| | |
|---|---|
| (Name) | Dialup |
| anchored | False |
| array | False |
| beginAnchored | False |
| beginDelim | |
| childMax | undefined |
| childMin | undefined |
| comment | |
| defaultValue | |
| defaultBytes | |
| defaultEncoding | ASCII |
| encoding | |
| endAnchored | False |
| endDelim | |
| endOfRec | False |
| fixedValue | |
| format | |
| inputMatch | |
| javaName | Dialup |
| javaType | java.lang.String |
| length | undefined |
| maxOccurs | 1 |
| member | |
| minOccurs | 1 |
| order | sequence |
| precedence | child |
| public | False |
| readOnly | False |
| reference | |

The above figure shows the layout of the Dial-up Event Type Definition file (dialup.xsc). There is a main node and two main child nodes. Currently the Dial-up e*Way only uses the DialupClient node. This version of Dial-up does not support the DialupServer node. This node is included expecting enhancements to be added.

The DialupClient node has the following child nodes and methods:

## Nodes in the XSC file:

- ConnectToModem
- DownloadFileByName
- ContentDownload
- UploadFileByName
- ContentUpload
- DisconnectModem

### ConnectToModem

Connects the Dial-up e*Way to the modem using the phone number specified. It also performs a login into the remote machine using the UserId, Password and MachineName fields.

The $1 node inside the ConnectToModem node indicates a repetition of the fields below it, but inside on ConnectToModem node the UserId, Password and MachineName fields occur only once. The order in which they are specified is left to the discretion of the user.

Every node under the DialupClient also has the BreakIfFail and OperationStatus fields.

- BreakIfFail: As the name indicates, if BreakIfFail is set to Yes, the file transfer file process is halted and exited. If it is set to No, the file transfer process continues, even if one node fails to perform the intended job. It is advisable that the user sets this field to Yes for the ConnectToModem node.
- OperationStatus: The OperationStatus field contains a string, which provides the user with the status of the work that the node is performing. This field indicates both a successful and an unsuccessful status. The user can check this field in the Collaboration Rule to monitor the results of the file transfer session. The possible values for OperationStatus are:
  - SUCCESS:Modem is intialized
  - SUCCESS:Modem is connected
  - FAILURE:Modem is not connected
  - FAILURE:File to upload already exists on remote machine, cannot proceed with Upload
  - FAILURE:Upload Directory does not exist on remote machine, cannot Upload.
  - SUCCESS:Kermit File upload is complete.

***Note:*** *Similar messages are also possible for x, y, and z modems.*

### DownloadFileByName

This node downloads a remote file to a local directory with the local file name given by the user. The remote file name is specified in the RemoteFile field, and the local file to download to is specified in the LocalFile field.

The remote file name can be provided with the directory path and name. The local file can be given with the directory path and name.

### ContentDownload

This node downloads a remote file and extracts the contents of that file. The contents are available to the user in the Data field of the node in the Collaboration Rule. This node can be used in both “Text” and “Binary” file transfer modes.

The data string available to the user is an encoded string using the ISO-8859-1 encoding format. In the Collaboration Editor Rule, the user can retrieve the specified string and write it to a file to recreate the desired file.

### UploadFileByName

This node uploads a local file to the remote machine with the remote file name given. The LocalFile field specifies the local file name.

### ContentUpload

This node uploads a data string to the remote machine with the remote file name provided. The content to be uploaded can be provided in the input Event file, or in the Collaboration Rule. This node can be used in both "Text" and "Binary" transer mode.

The e*Way expects a string to be encoded in ISO-8859-1 format as the content to upload. The e*Way writes the string to the remote file name specified and then uploads the file.

### DiconnectModem

This node disconnects the modem connection and logs out of the remote machine. If this node is not specified in the input Event the modem disconnects and logs out of the remote machine when the e*Way is shut down.

### Methods in the XSC file

- process

The 0$ node in the dial-up input Event Definition contains node repetitions and indicates that the nodes of DialupClient can occur any number of times in the input Event. The instance of $0 is a vector where each element can be any one of the above-mentioned nodes. The first node in the $0 vector starts at an index of 0. The process method will iterate through the instance of the $0 vector and look at each node, identify the type of that node and then call the appropriate method to perform the required operation. Thus when the user is specifying the input Event he/she to provide all the necessary parameters, which will be needed for the file transfer process to be carried out by the process method.

As mentioned before he can specify the necessary parameters in the input Event or in the Collaboration Rules.

For example, when the Auto Connect Flag is set to No then in the input Event the user needs to give the ConnectToModem node first and then any of the upload or download nodes in any sequence he may prefer.

The process method returns a Boolean value of "true" or "false" to indicate the success or failure of the process operation. If all the nodes of the input Event are processed successfully, then process returns "true". If any of the nodes fail at the intended function, process returns "false". In the Collaboration Editor, the user can check the status of the return value from process, and determine which of the nodes failed by adding the following rule after the process.

```
if(getOut().getDialupClient().get$0(index of node
).getNODE_NAME().getOperationStatus().indexOf("F")  == 0){
System.out.println("Op status is - " +
getOut().getDialupClient().get$0(index of node
).getNODE_NAME().getOperationStatus());
}
```

The “index of node” is the index of the node whose status is being checked. For NODE_NAME in getNODE_NAME(), substitue the name of the node, whose operation status you are checking. The check for “F” is made because the operation status field has a value of “SUCCESS” or “FAILURE” before the status text message. The user can thus look for the start of the alphabet to determine if the node’s execution was a SUCCESS or FAILURE.

## Sample Dial-up Input Event XML File:

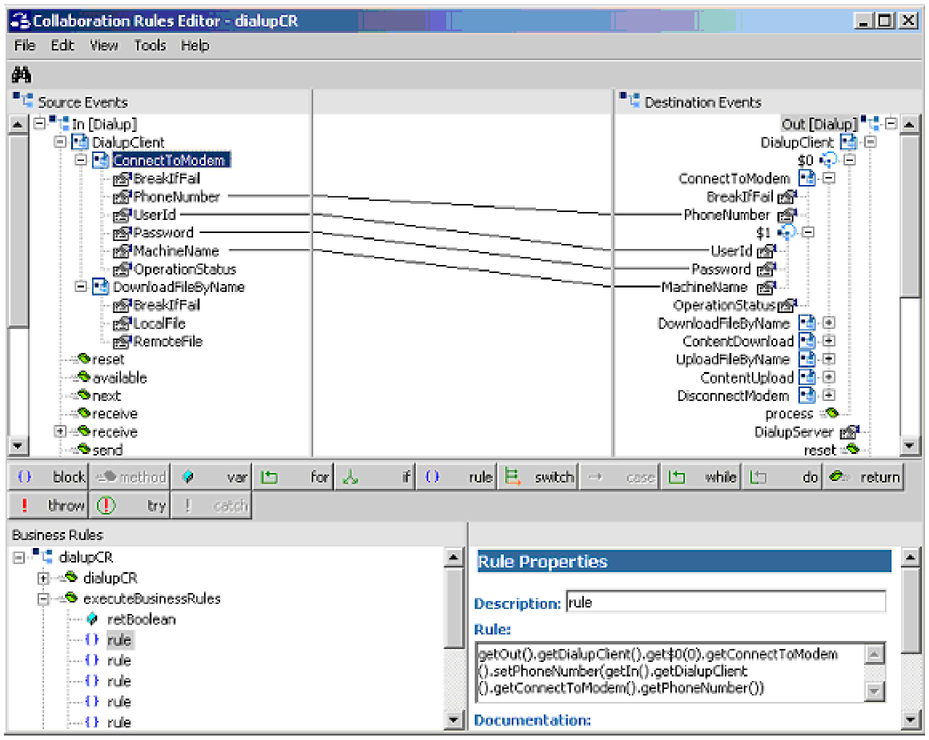
A sample input Event in XML format is provided below:

```
<Dialup>
<DialupClient>
    <ConnectToModem BreakIfFail="yes">
        <PhoneNumber>6264716197</PhoneNumber>
        <UserId>seeBeyond</UserId>
        <Password>eai</Password>
        <MachineName>atlas</MachineName>
        <OperationStatus>status</OperationStatus>
    </ConnectToModem>
    <UploadFileByName BreakIfFail="yes">
        <LocalFile>C:\INDATA\dialtest.txt</LocalFile>
        <RemoteFile>/upload/temp/dialtest.txt</RemoteFile>
        <OperationStatus>status</OperationStatus>
    </UploadFileByName>
    <ContentUpload BreakIfFail="yes">
        <Data>this is a dialup test for contentupload</Data>
        <RemoteFile>/upload/data.txt</RemoteFile>
        <OperationStatus>status</OperationStatus>
    </ContentUpload>
    <DownloadFileByName BreakIfFail="yes">
        <LocalFile>C:\INDATA\qa_29885.pl</LocalFile>
        <RemoteFile>qa_29885.pl</RemoteFile>
        <OperationStatus>status</OperationStatus>
    </DownloadFileByName>
    <ContentDownload BreakIfFail="yes">
        <Data>""</Data>
        <RemoteFile>/download/setting.txt</RemoteFile>
        <OperationStatus>status</OperationStatus>
    </ContentDownload>
    <DisconnectModem BreakIfFail="yes">
        <OperationStatus>test</OperationStatus>
    </DisconnectModem>
</DialupClient>
</Dialup>
```

***Note:*** *The user need not use this XML file for his input event. He can create his own input Event Definition File and corresponding XSC file. In this case he will have to map the fields in his custom input file to the fields of the Dialup Event type Definition in the Java Collaboration Editor.*

The figure below shows the mapping of the input Event Type Definition to the dialup Event Type Definition using the Java Collaboration editor.

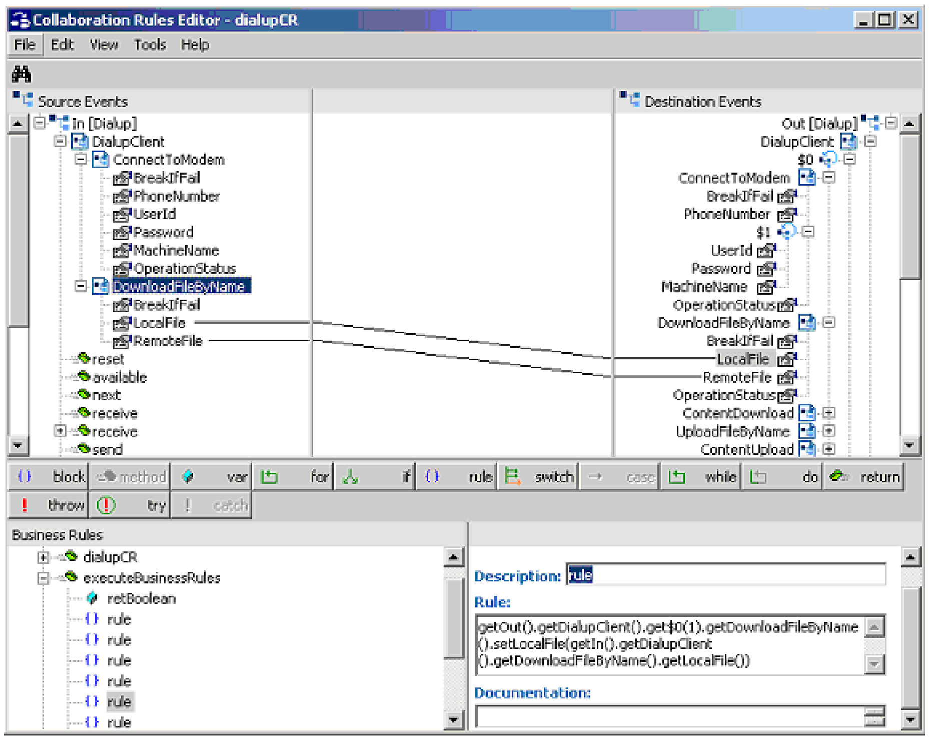
**Figure 2** Sample Dialup XSC File



Description: rule

```
getOut().getDialupClient().get$0(0).getConnectToModem
().setPhoneNumber(getIn().getDialupClient
().getConnectToModem().getPhoneNumber())
```

**Figure 3** Sample Dialup XSC Mapping



Description: rule

```
getOut().getDialupClient().get$0(1).getDownloadFileByName
().setLocalFile(getIn().getDialupClient
().getDownloadFileByName().getLocalFile())
```

After the required and necessary mapping is done the user needs to call the "process" method of the dialup.xsc. This method will process all the nodes and start the file transfer operation.

Any status check or monitoring of the file transfer operation can be done after the process method by checking the values of the OperationStatus fields in the dialup event type definition.

## Requirements in Specifying File Names in the Input Event

For downloads and uploads on Windows systems, you must give a full qualified path name of the files such as "C:\INDATA\temp.txt", as there can be many drives on a Windows machine with different write access. If the directory name is not specified the local file will be placed in the same directory as the input event file given in the feeder configuration.

For uploads and downloads to a UNIX machine, the user can specify a relative path on the machine such as "/upload/downdloadDoc.txt". This path is relative to the home directory of the user with whose userid the e*Way is logged in the remote machine.

Thus the following path specifications needs to be followed in the input data file.

### File Upload

**Windows -> UNIX**

Local File: Full Path: C:\Temp\filename.txt

Remote File : Relative Path: /upload/FileToUpload.txt

**UNIX -> UNIX**

Local File: Full Path: /home/userId/fileToUpload.txt

Remote File: Relative Path: /upload/UploadedFile.txt

### File Download

**UNIX -> Windows**

Local File: Full Path: C:\Temp\DownloadedFile.txt

Remote File: Relative Path: /upload/filetoDownload.txt

**UNIX -> UNIX**

Local File: Full Path: /home/userId/DownloadedFile.txt

Remote File: Relative Path: /upload/FileToDownload.txt

### 4.1.2 X/Y Mode Protocol Specifications

X and Y modems send data in 128 byte "sectors" (terminology from X Modem's CP/M origins). X Modem does not provide the receiving end with any indication of how long the file actually is. The receiving end does no know how many bytes of the last sector belongs to the file. The receiving end writes all 128 bytes of the last sector to the end of the file. The received file always has a size that is a multiple of 128 bytes. Unless the original files size is a multiple of 128 bytes, a file received using X/Y Modem has extra characters at the end. It is important to use a program, such as zip or tar to "package" the files before transferring them with X Modem. These programs can handle garbage characters at the end of the file. Therefore, it is very important to package your binary files before you transfer them with X or Y modem.

### 4.1.3 Installing the Sample Schema

The sample schema is located on the Installation CD in the following folder:

```
\samples\ewdialup
```

For more information on importing sample schemas, see Chapter 6, Migrating Schemas and Components of the *e*Gate Integrator System Administration and Operations Guide*.

### 4.1.4 Execute the Schema

To execute the Dial-up sample schema, do the following:

1. Go to the command line prompt, and enter the following:

   ```
   stccb -rh hostname -rs dialupSample -un username -up user password
   -ln hostname_cb
   ```

   Substitute ***hostname***, ***username*** and ***user password*** as appropriate.

2. Exit from the command line prompt, and start the Schema Manager GUI.

3. When prompted, specify the ***hostname*** which contains the Control Broker you started in Step 1 above.

4. Select the dialupSchema schema.

5. After you verify that the Control Broker is connected (the message in the Control tab of the console will indicate command *succeeded* and status as *up*), highlight the IQ Manager, ***hostname***_igmgr, then click on the right button of the mouse, and select **Start**.

6. Select each of the e*Ways, right-click the mouse, and select **Start**.

***Note:*** *While the schema is running, opening the destination file, will cause errors.*

#### Troubleshooting

Once the connection is established, and the user has logged in successfully, if an abnormal termination is encountered, (i.e., killing the CB, or telephone line is dropperd) the csh process generated by the e*Way continues to run on the remote host. The user should clean up these processes (i.e., kill -9).

# Index