*e*Index Global Identifier Product Suite*

e*Index Initial Load User's Guide

Release 5.0.5 for Schema Run-time Environment (SRE)

SEEBEYOND

The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

SeeBeyond, e*Gate, e*Way, and e*Xchange are the registered trademarks of SeeBeyond Technology Corporation in the United States and/or select foreign countries. The SeeBeyond logo, SeeBeyond Integrated Composite Application Network Suite, eGate, eWay, eInsight, eVision, eXchange, eView, eIndex, eTL, ePortal, eBAM, and e*Insight are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2005 SeeBeyond Technology Corporation. All Rights Reserved.

This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.

Version 20050401124200

Table of Contents

Chapter 1: Introduction 1-1
About this Chapter 1-1
Overview 1-1
What's Inside 1-2
Introduction 1-3
Welcome 1-3
What is the Scope of this Guide? 1-3
Who Should Use this Guide? 1-3
How Should this Guide be Used? 1-4
How is this Guide Organized? 1-4
What Conventions are Used in this Guide? 1-5
Learning About the Initial Load Program 1-7
Overview 1-7
About the Initial Load Process 1-7
About the Schemas 1-7
About the Validation Phase 1-7
About the Load Phase 1-8
About Reports 1-8
Initial Load Overview 1-8
Additional Resources 1-11
Chapter 2: Installing the Initial Load Program 2-1
About this Chapter 2-1
Overview 2-1
What's Inside 2-2
Learning About the Initial Load e*Ways 2-3
Overview 2-3
Getting Started 2-3
Validation Schema 2-3
Load Schema 2-4
Installing the Initial Load e*Ways 2-5
Overview 2-5
Step 1: Back up the e*Gate Environment 2-5
Step 2: Install or Upgrade e*Gate 2-5
Step 3: Install or Upgrade the Database e*Way 2-6
Step 4: Install Database Client Software 2-6
Step 5: Install the e*Index Schema Components 2-6
Step 6: Install the e*Index Database 2-7
Step 7: Copy the Initial Load Program Files 2-7
Step 8: Modify the Installation Script 2-7
Step 9: Run the Installation Script 2-8
Step 10: Verify the Schema Directory Structure 2-10
Step 11: Create the Data Directory Structure 2-11
Step 12: Delete Extraneous Files 2-11
Chapter 3: Validating the Input Data 3-1
About this Chapter 3-1
Overview 3-1
What's Inside 3-2
About the Validation Process 3-3
Overview 3-3
Requirements 3-3
About the Validation Phase 3-3

Validation Schema Architecture 3-4
Validation Schema Customizations 3-7
Validation Functions 3-9
Valid Value Lists 3-11
Exclusion Lists 3-12
Important Validation Information! 3-13
Output Files 3-13
Data File Requirements 3-14
Overview 3-14
Formatting Guidelines 3-14
Sample Inbound ETD 3-20
Important Information About Dates and Times of Events 3-20
Standardizing the Data File 3-21
Overview 3-21
Step 1: Obtain the Data File and Lists 3-21
Step 2: Modify ui-conv-lists.monk 3-21
Step 3: Modify ui-check-functions.monk 3-23
Step 4: Modify ui-custom.monk 3-23
Step 5: Modify the ETDs 3-24
Step 6: Modify the Collaboration Script 3-25
Step 7: Copy the Input Data Files 3-26
Step 8: Configure the e*Ways 3-27
Step 9: Validate the Data 3-28
Step 10: Analyze the Data Output 3-29
Chapter 4: Loading the Validated Data 4-1
About this Chapter 4-1
Overview 4-1
What's Inside 4-2
Learning About the Load Schema 4-3
Overview 4-3
Requirements 4-3
About the Load Phase 4-3
Output Files 4-3
Load Schema Architecture 4-4
Load Schema Customizations 4-6
Dropping Indexes 4-7
A Note About the Audit Trail 4-8
Loading the Data File 4-9
Overview 4-9
Step 1: Copy the Data Files 4-9
Step 2: Configure the Database 4-10
Step 3: Modify the Monk Files 4-11
Step 4: Modify the ETDs 4-11
Step 5: Modify the Collaboration Script 4-12
Step 6: Configure the e*Ways 4-12
Step 7: Stop the Archive Log 4-14
Step 8: Load the Data 4-14
Step 9: Fix Records in the Error File 4-15
Step 10: Restart the Archive Log 4-15
Step 11: Run Database Reports 4-16

Chapter 1

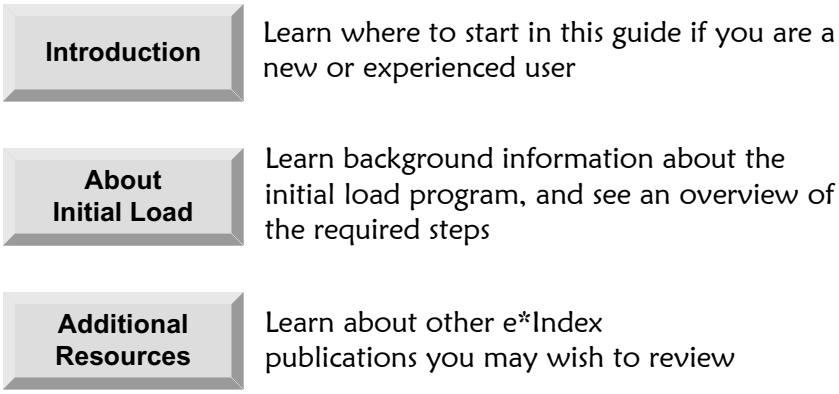
Introduction

About this Chapter

Overview

This introduction provides an overview of the steps you need to follow to load existing data from your legacy systems into an e*Index™ Global Identifier database.

The following diagram illustrates the contents of each major topic in this chapter. For the page numbers on which specific topics appear, see "What's Inside" on the following page.



What's Inside

This chapter provides background information and instructions related to the topics listed below.

Introduction 1-3
Learning About the Initial Load Program 1-7
Additional Resources 1-10

Introduction

Welcome

Welcome to e*Index, SeeBeyond's enterprise-wide master person index. This document explains how to load your existing legacy data into the e*Index database. This chapter of the document provides a summary of the process, as well as information you should know before beginning the initial load procedure.

Whether you are a new or established user, you should read through this guide before you begin the load process. Please pay particular attention to the overview sections provided at the beginning of each chapter and at the beginning of each section within a chapter. The overview sections are designed to provide background and explanatory information you may need to understand. After reading the overview information, you will be ready to perform specific tasks using the step-by-step instructions provided in each chapter.

What is the Scope of this Guide?

This guide provides background information and step-by-step instructions for loading existing legacy data in the e*Index database. It includes functional instructions, and background information where required. This guide provides information about installing the initial load program, configuring the e*Way files, and the required format of the data file.

This guide does not include information or instructions on using any of the e*Index applications or the e*Index schema. These topics are covered in the appropriate user's guide (for more information, see "Additional Resources" on page 1-7).

Who Should Use this Guide?

Any user who will load legacy data into the e*Index database should read this guide. Any user who will create the data extract file containing the legacy data should be sure to read "Formatting Guidelines" in Chapter 3 of this guide to learn about the desired format of the file.

A thorough knowledge of e*Index is not needed to understand this guide. It is presumed that the reader of this guide is familiar with the e*Gate environment, e*Gate schemas, and database administration, and the operating system(s) on which e*Gate and the e*Index database run.

How Should this Guide be Used?

For best results, you should skim through the guide to familiarize yourself with the locations of essential procedures you need to perform. Each chapter begins with a simple graphic that identifies the information contained in the chapter. The second page of each chapter contains a list of topics and instructions included in the chapter and the associated page numbers.

How is this Guide Organized?

This guide is divided into four chapters that cover the topics shown below.

Chapter	Topics
Chapter 1, Introduction	■ Introduction ■ About the Initial Load Program ■ Additional Resources
Chapter 2, Installing the Initial Load Program	■ About the Initial Load Program ■ Performing the Installation
Chapter 3, Validating the Data	■ About the Validation Process ■ About the Data Extract File ■ Performing the Validation
Chapter 4, Loading the Data	■ About the Load Process ■ Performing the Initial Load

What Conventions are Used in this Guide?

Before you start using this guide, it is important to understand the icon, special notation, and mouse conventions used.

Icon and Special Notation Conventions

The following conventions are used in this and other e*Index publications to identify special types of information.

Icon or Notation	Type of information
Note	Supplemental information that is helpful to know, but not essential to completing a particular task.
Tip	Information that helps you to apply techniques and procedures described in the text to your specific needs. May also suggest alternative methods.
Important!	Information that is essential to the completion of a task.
Caution!	Advises you to take specific action to avoid loss of data.
	Indicates the beginning of a step-by-step instruction.
	Specifies a task to perform before you begin a step-by-step instruction.
	Indicates a cross-reference to other sections of the guide or to other publications.

Mouse Conventions

You can use either a single-button mouse or a multiple-button mouse with e*Index. If you use a multiple-button mouse, the left mouse button is the primary button, unless the mouse is configured differently.

The instructions in this guide may require you to use the mouse in a variety of ways:

- **Point** means to position the mouse pointer until the tip of the pointer rests on whatever you want to point to on the screen.
- **Click** means to press and then immediately release the left mouse button without moving the mouse.
- **Double-click** means to click the left mouse button twice, in rapid succession.
- **Right-click** means to click the right mouse button once.
- **Drag** means to point and then hold down the mouse button as you move the mouse. **Drop** means to let go of the mouse button to place the dragged information where you want it to be moved.
- **Move** means to point to an object on the screen, such as an e*Index Security user group, and drag the mouse to move the object to a screen location of your choice.
- **Highlight** means to select an area of text by dragging the mouse over the desired portion of text that appears on a window.
- **Select** means to point to a list of information on an e*Index window, and then click once to choose the data you want. The information becomes highlighted when selected.
- **Expand** means to double-click a row of information on an expandable list to display more details. The details appear on another row, below the row you double-click.
- **Collapse** means to double-click a row of information on an expandable list to hide the details that appear on the following row.

Learning About the Initial Load Program

Overview

This section of the chapter provides background information about the initial load program and a summary of the load process.

About the Initial Load Process

The initial load program consists of two e*Gate schemas that you run in two phases: 1) validate the data and 2) load the data into the database. If the data extract file is not provided in the required format, or if you need to validate certain data elements, you must validate the data file before loading it into the database. You perform the data validation using a validation schema that you customize according to the data extract file. Once you have completed the validation, the data file is ready to be loaded. The data load is performed using a different e*Gate schema, which is designed for multi-threaded execution and runs several e*Ways simultaneously.

About the Schemas

The initial load program is provided in two separate e*Gate schemas, which you will need to customize for your business requirements. You perform the installation using a simple batch file that registers the schemas with the registry automatically. You will need to modify this batch file slightly for your e*Gate environment.

The initial load schemas are designed to be a template that should be customized to meet the individual requirements of each site. Most of the customization will occur in the validation schema. There is no direct interaction between the two schemas.

About the Validation Phase

Phase 1 is the validation stage, and it transforms the input file into the required format. This format is described in Chapter 3 of this guide, "Validating the Input Data". To perform this phase of the load process, you need to make some customizations to the schema provided. This part of the process uses valid value lists, which define the values allowed in specific fields of the data extract file. It also uses exclusion lists, which define the values *not* allowed in specific fields of the data extract file. This phase also performs any additional validations that you specify against the data extract file.

This validation phase produces several output files that you can use to analyze the state of the data in the extract file, and to correct any errors in the

data or in the validations. Use this phase of the load process to ensure that the data you are loading conforms to your business requirements and to the requirements of e*Index.

About the Load Phase

Once you complete the validation phase against the data extract file, the data is ready for you to load into the database. During the load process, you can monitor the progress of the data load by checking a counter in the database, or by reviewing the log files. Depending on the size of the database and the number of records to process, this may take several days. Any records that cannot be processed due to errors are written to an error file. When the load process is complete, you can fix the records written to this file, and then rerun the program against the error file to ensure that all records in the data file are processed.

About Reports

After you have completed the initial load process, you should run standard reports against the database to check for potential duplicate records, for default values in the loaded records, and so on. For more information about running these reports, see *Working with Reports for e*Index Global Identifier*.

Initial Load Overview

This document provides instructions for installing the initial load program, and for running the validation and load phases of the program. Each of these tasks is described in its own chapter of this guide. For a summary of the steps that you need to complete for the initial load, see the diagram on page 1-10.

The steps for performing the initial load are:

- Step 1: Install the Initial Load Program (described in Chapter 2)
- Step 2: Validate the Data (described in Chapter 3)
- Step 3: Load the Data (described in Chapter 4)

Step 1: Install the Initial Load Program

Chapter 2, "Installing the Initial Load Program", outlines the steps required to install the schema files into your e*Gate environment. This chapter also discusses the components installed into each schema of the initial load program.

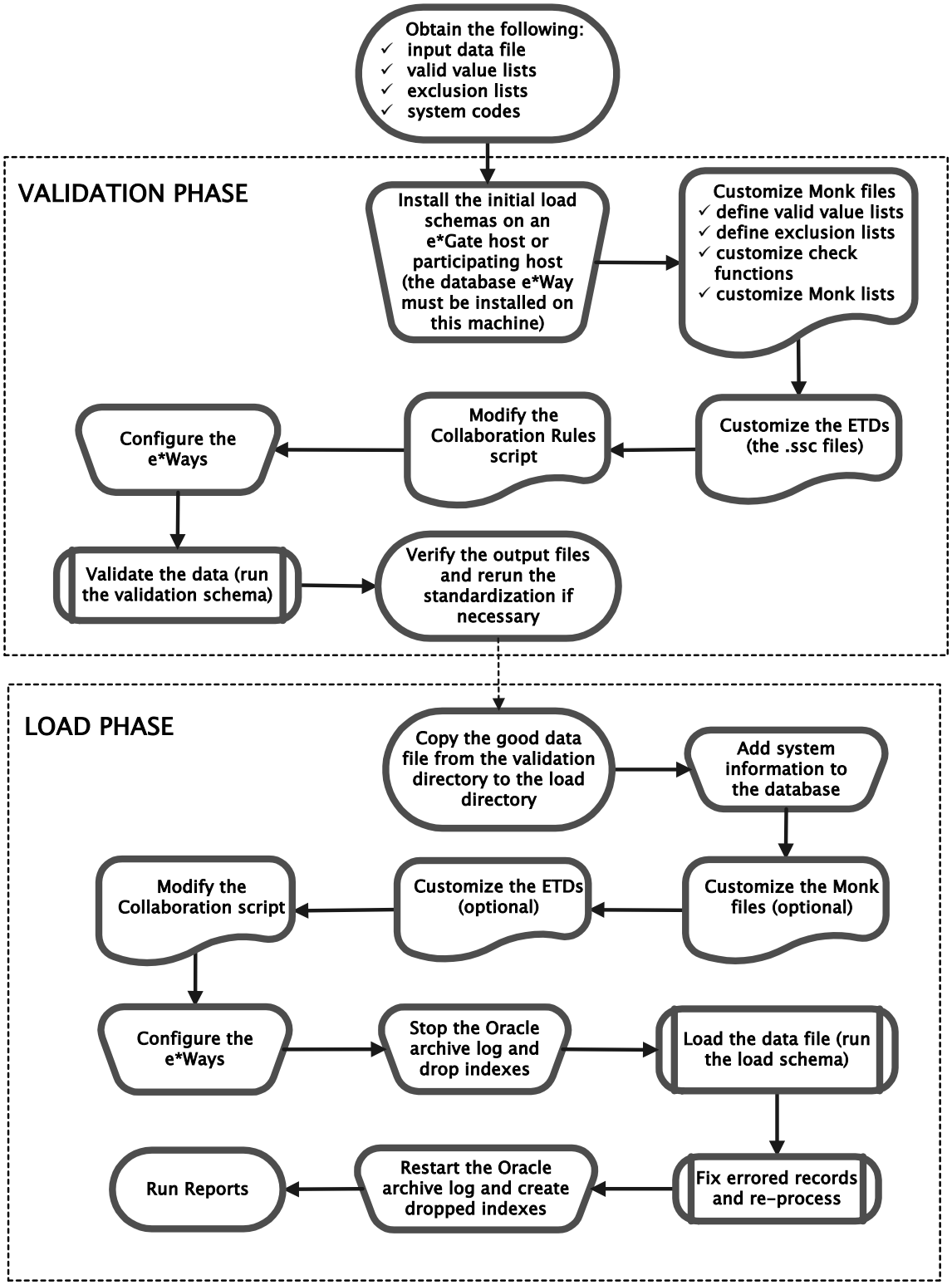
Step 2: Validate the Data

Follow the steps in Chapter 3, "Validating the Data", to reformat the data extract file into a format that is acceptable by e*Index and that conforms to your business requirements. This chapter includes descriptions of the valid value and exclusion lists you can use, available validation commands, output files, and a thorough description of the format to which the data extract file must conform before being loaded into the e*Index database. You perform most of the site-specific customizations during this phase.

Step 3: Load the Data

Chapter 4, "Loading the Data", contains the instructions for completing the initial load process by loading the data into the e*Index database. This chapter also provides information about the files produced during the load process, information that you must add to the database prior to initiating the load process, monitoring the load process, and correcting any errors that occur. Before you begin this process, you must have an e*Index 5.0.5 for SRE database in place.

Figure 1-1: Initial Load Overview



Additional Resources

SeeBeyond has developed a suite of e*Index user's guides and related publications that are distributed in an electronic library.

- *e*Index Global Identifier User's Guide*
  Helps e*Index quality workstation users to perform database maintenance tasks, such as merging and unmerging records, finding and resolving potential duplicates, adding and updating records, and viewing the audit trail.
- *e*Index Administrator User's Guide*
  Helps system administrators configure system parameters, customize e*Index, work with Vality rule set files, and processing codes. This guide also describes how to maintain the information in the database that is used to populate the drop-down lists in the e*Index.
- *e*Index Security User's Guide*
  Helps system administrators add users and user groups to e*Index, to grant security permissions to users and user groups, to maintain user and user group information, and to configure certain system parameters.
- *e*Index Global Identifier Technical Reference*
  Describes message processing for e*Index, as well as database tables and e*Index Monk APIs. This guide also provides a complete listing of e*Index Monk APIs and functions, along with a description, parameters, syntax, return values, and examples for each.
- *Working with Reports for e*Index Global Identifier*
  Provides background information about the GUI and standard reports provided with e*Index, and explains how to modify and run the standard reports (for an Oracle installation only).
- *e*Index Global Identifier Installation Guide*
  Helps system and database administrators install a new e*Index environment for the current release, including e*Index schema files, the e*Index GUI, and database installation.
- *e*Index Global Identifier Upgrade Guide*
  Helps system and database administrators upgrade an existing e*Index environment to the most current release, including e*Index schema files, the e*Index GUI, and database upgrades.
- *Java Programmer's Guide for e*Index Active Integration*
  Provides background and implementation information about the Java APIs for e*Index Active Integration. This guide also provides a complete listing of e*Index Java functions, along with a description, parameters, syntax, return values, and examples for each.

Chapter 2

Installing the Initial Load Program

About this Chapter

Overview

This chapter presents the background information and the step-by-step instructions you need to install the e*Gate schema files for the e*Index initial load program.

The following diagram illustrates the contents of each major topic in this chapter. For the page numbers on which specific topics appear, see the next page of this chapter.

About Initial Load Schemas — Learn about the installation process, the requirements, and the files you will be installing

Install the Schemas — Learn how to install the initial load files in your e*Gate environment

What's Inside

This chapter provides background information and instructions related to the topics listed below.

Learning about the initial load e*Ways 2-3
Installing the initial load e*Ways 2-5
- Step 1: Back up the e*Gate Environment 2-5
- Step 2: Install or Upgrade e*Gate 2-5
- Step 3: Install or Upgrade the Database e*Way 2-6
- Step 4: Install Database Client Software 2-6
- Step 5: Install the e*Index Schema Components 2-6
- Step 6: Install the e*Index Database 2-7
- Step 7: Copy the initial load program files 2-7
- Step 8: Modify the installation script 2-7
- Step 9: Run the installation script 2-8
- Step 10: Verify the Schema Directory Structure 2-10
- Step 11: Create the Data Directory Structure 2-11
- Step 12: Delete Extraneous Files 2-11

Learning About the Initial Load e*Ways

Overview

This section of the chapter provides background information about installing the e*Gate schema files for the initial load program, and the files that are installed for each schema.

Getting Started

Installing the initial load may require some modifications to your e*Gate and Database e*Way environments. We highly recommend that a separate e*Gate and Database e*Way environment be created and tested before any modifications are made to your current working environment. Before you start you will need to have the following software from SeeBeyond for the platform with which you are working.

- e*Gate 5.0.5 for SRE
- The appropriate Database e*Way 5.0.5 for SRE
  - For an Oracle database, you need the Oracle e*Way
  - For a Sybase database, you need the Sybase e*Way
  - For a Microsoft SQL Server database, you need the ODBC e*Way
- HL7 Templates 5.0.5 for SRE (only if you will be processing HL7 messages)
- e*Index 5.0.5 for SRE (database and e*Gate schema components only)

You also need the appropriate database client software installed on the initial load machine. You do not need to have the e*Index GUIs installed at the time you perform the initial load.

Validation Schema

The batch file installs the validation schema files in the registry of the e*Gate server in **/<eGate>/Server/Registry/repository/ei_validate/runtime**. The schema also uses files from the e*Index schema components, which are located in **/<eGate>/Server/Registry/repository/default**. The diagram on page 2-10 illustrates the directory structure for the validation and load schemas.

The validation schema consists of four event types with their associated event type definitions (ETDs), one BOB, eight e*Ways (two sets of four for simultaneous processing), and the Intelligent Queue (IQ). This schema also includes a Monk library of **.monk** files used for defining the validations and

check functions for the schema. The associated files are all installed in the **../ei_validate/runtime** directory.

Load Schema

The batch file installs the load schema files in the registry of the e*Gate server in **/<eGate>/Server/Registry/repository/ei_load/runtime**. The schema also uses files from the e*Index schema components, which are located in **/<eGate>/Server/Registry/repository/default**. The diagram on page 2-10 illustrates the directory structure for the validation and load schemas.

The load schema consists of two event types with their associated event type definitions (ETDs), seven e*Ways (using subscriber pooling), and the IQ. This schema also includes a Monk library of **.monk** files used to define database connectivity functions and processing rules for the schema.

Installing the Initial Load e*Ways

Overview

To install the schemas for the initial load program, you must complete the following steps:

- Step 1: Back up the e*Gate Environment
- Step 2: Install or Upgrade e*Gate
- Step 3: Install or Upgrade the Database e*Way
- Step 4: Install Database Client Software
- Step 5: Install the e*Index Schema Components
- Step 6: Install the e*Index Database
- Step 7: Copy the Initial Load Program Files
- Step 8: Modify the Installation Script
- Step 9: Run the Installation Script
- Step 10: Verify the Schema Directory Structure
- Step 11: Create the Data Directory Structure
- Step 12: Delete Extraneous Files

Note: *If you have already installed the e*Index database and the e*Gate schema files for e*Index, you have already completed steps 1 through 6. You can begin at "Step 7: Copy the Initial Load Program Files.*

Step 1: Back up the e*Gate Environment

Before making any changes to your e*Gate environment, it is important to make a FULL backup of the environment for safekeeping.

Step 2: Install or Upgrade e*Gate

Before you can run the initial load schemas, you need to either install or upgrade to e*Gate 5.0.5 for SRE. Follow the instructions for installing or upgrading e*Gate, which can be found in your *e*Gate Integrator Installation Guide*. If you are processing HL7 messages, make sure you also install the HL7 template libraries add-on for e*Gate. If you are not currently using e*Gate 5.0.5 for SRE, it should first be installed on a separate environment for testing purposes, away from your production e*Gate environment.

Step 3: Install or Upgrade the Database e*Way

Before running the initial load schemas, you need to install or upgrade the appropriate Database e*Way (version 5.0.5 for SRE) for your database platform. For an Oracle database, install the Oracle e*Way; for a Sybase database, install the Sybase e*Way; and for a SQL Server database, install the ODBC e*Way. For more information on installing the Database e*Way, refer to the user's guide for that e*Way.

Step 4: Install Database Client Software

In order for the load schema to connect with the e*Index database, you need to install the appropriate database client software on the e*Gate host on which the load schema is installed. This may be the registry host or a participating host. Before you install the software, make sure that no previous versions exist. You need to install one of the following:

- For an Oracle database, install Oracle Client 8.1.7 or 9*i*, including the Oracle network administration and application development tools. After you install Oracle, modify **tnsnames.ora** by adding a stanza for the new e*Index database (for more information, see "Step 11: Modify **tnsnames.ora**" in Chapter 3 of the *e*Index Global Identifier Installation Guide*).
- For a Sybase database, install Sybase Client 12.0. After you install Sybase, modify **sql.ini** by adding a stanza for the e*Index database server (for more information, see "Step 12: Modify **sql.ini**" in Chapter 4 of the *e*Index Global Identifier Installation Guide*).
- For a Microsoft SQL Server database, install Microsoft SQL Server Enterprise Edition 7.0. You only need the client files for the e*Gate server. After you install SQL Server, you need to define an ODBC data source for the database (see "Step 8: Define the ODBC Data Source" in chapter 5 of the *e*Index Global Identifier Installation Guide*).

For more information about installing the database software, refer to the appropriate Oracle, Sybase, or Microsoft SQL Server documentation.

Step 5: Install the e*Index Schema Components

The initial load program relies on several of the e*Index schema components in order to function properly. Before you copy the initial load program files, make sure you install the schema components on the e*Gate host on which you will be installing the initial load program files. For more information about installing the schema components, see Chapter 2, "Installing the e*Gate Schema Files", in the *e*Index Global Identifier Installation Guide*. You do not need to install the sample schema in order to run the initial load.

Step 6: Install the e*Index Database

You need to create an e*Index database prior to performing the initial load. Make sure the database is installed on a database server, and is started and accessible before performing the load. For more information about installing the database, see the appropriate database installation chapter of the *e*Index Global Identifier Installation Guide*.

Step 7: Copy the Initial Load Program Files

The initial load program files are distributed in a compressed .TAR file that you need to copy from the e*Index CD-ROM. You then need to extract the .TAR file. Regardless of the platform on which the e*Gate host runs, the initial load programs need to be extracted on a Windows machine.

Note: *If you still have the files that were installed when you installed the e*Index schema components, then you can skip this step. You can find the extracted initial load files in the path you specified for the installation in the* ***/client/eGate_schema/init_load*** *subdirectory. Skip to "Step 8: Modify the Installation Script".*

▶ To copy the initial load program files

Before you begin:

- ✓ Complete "Step 1: Backup your e*Gate Environment" through "Step 6: Install the e*Index Database"

1. Insert the e*Index installation CD-ROM into the CD-ROM drive of your computer.
2. On the Windows desktop, double-click **My Computer**, open the CD-ROM directory, and then navigate to the **init_load** directory.
3. Copy the file **init_load.tar** to a temporary directory.
4. Extract the file **init_load.tar** to a temporary directory using WinZip. Be sure to select **Use Folder Names** in the lower left portion of the Extract dialog.
5. Continue to "Step 8: Modify the Installation Script."

Step 8: Modify the Installation Script

SeeBeyond provides an installation script for the initial load schemas that will automatically copy the files to the e*Gate registry and commit the new schemas. Before you run this script, you need to specify certain information, such as the registry host name and the location of the e*Gate directories.

▶ To modify the installation script

Before you begin:

✓ Complete "Step 7: Copy the Initial Load Program Files"

1 Navigate to the directory into which you extracted the initial load program files, and open the file **install.bat** in any text editor. Do not double-click the file to open it.

2 If you already have initial load schemas installed on the e*Gate host, and you do not want to lose any customizations, you need to change the schema names for the validate and load schemas. To do so, modify the following lines.

```
set VALIDATION_SCHEMA_NAME=ei_validate

set LOAD_SCHEMA_NAME=ei_load
```

3 Modify the following line by replacing **localhost** with the name of your registry host server.

```
set REGISTRY_HOST=localhost
```

The modified line would look something like this:

```
set REGISTRY_HOST=IBM-437
```

Note: *You can also modify login ID and password information for the system administrator and the name of the installation log file. This is not required.*

4 Save and close **install.bat**.

5 Continue to "Step 9: Run the Installation Script."

Step 9: Run the Installation Script

Once you have modified the installation script, you can run the script to install the new schema files and commit them to your e*Gate registry service.

▶ To run the installation script

Before you begin:

✓ Complete "Step 8: Modify the Installation Script"

1 Navigate to the directory into which you extracted the initial load program files.

2 Do one of the following:

To run the file from My Computer or Windows Explorer, double-click on the file **install.bat**.

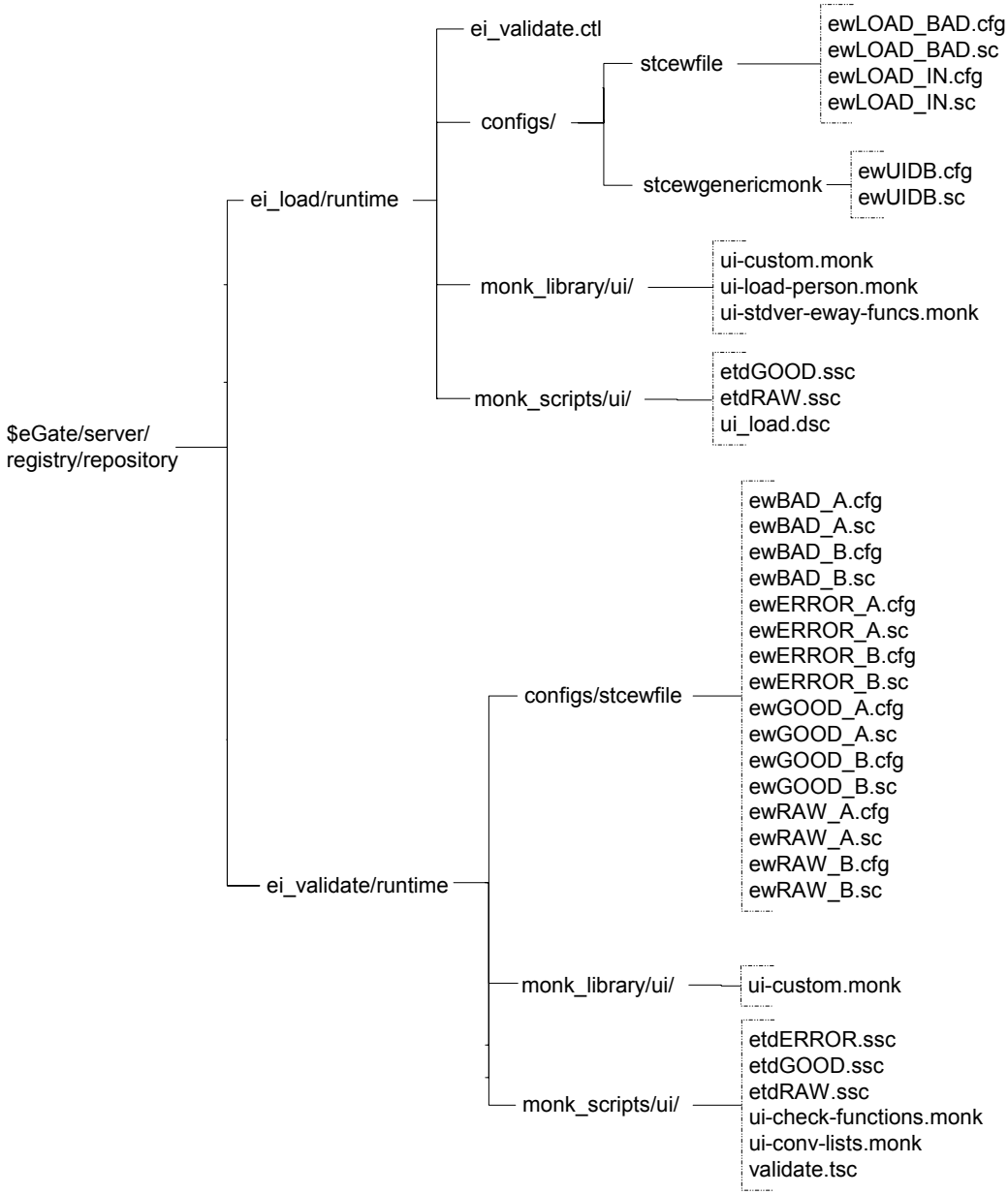
To run this file from an MS-DOS command line, open the command line, navigate to the directory in which **install.bat** is located, and then type **install**.

3 Review the log file for any errors. By default, this file is named **install.log** and is located in the directory from which you ran the installation script.

4 Continue to "Step 10: Verify the Schema File Structure"

Step 10: Verify the Schema Directory Structure

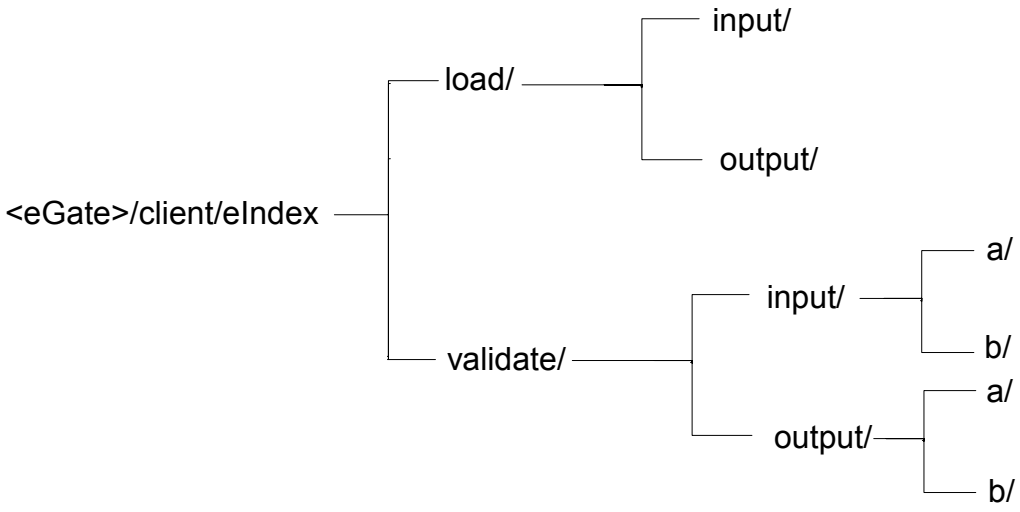
After you install the schema files, you can use the following diagram to verify and familiarize yourself with the files installed into each schema directory.



Step 11: Create the Data Directory Structure

If you want to use the default values for the location of the input and output files for the load process, you need to create the directories in your e*Gate home directory. By default, the e*Ways for the validation schema are set up to locate source files and print output files in **/<eGate>/client/eIndex/validate**. The e*Ways for the load schema are set up to locate source files and print output files in **/<eGate>/client/eIndex/load**. You do not have to use the specified directories, but you need to modify the e*Way configuration files if you use different paths.

The diagram below illustrates the directory structure expected by the default e*Ways for the input and output files for both schemas. In this diagram, the input files processed for each phase go in the **input** directories, and the e*Ways write the output files to the **output** directories. This is explained in more detail in the following chapters.



Step 12: Delete Extraneous Files

Once you have successfully installed the initial load schemas, you can delete the initial load files from the temporary directory into which you originally extracted them.

Chapter 3

Validating the Input Data

About this Chapter

Overview

This chapter presents the background information and the step-by-step instructions you need to validate and standardize legacy data before you load it into the e*Index database.

The following diagram illustrates the contents of each major topic in this chapter. For the page numbers on which specific topics appear, see the next page of this chapter.

The Validation Process	Learn about the validation schema, it's components, and it's design
The Data File	Learn about the default Event Type Definition and required format for the input data file
Validating the Data	Learn the step-by-step instructions for customizing and running the validation schema

What's Inside

This chapter provides background information and instructions related to the topics listed below.

About the Validation Process 2-3
Data File Requirements 2-14
Standardizing the Data File 2-21
- Step 1: Obtain the Data File and Lists 2-21
- Step 2: Modify **ui-conv-lists.monk** 2-21
- Step 3: Modify **ui-check-functions.monk** 2-23
- Step 4: Modify **ui-custom.monk** 2-23
- Step 5: Modify the ETDs 2-24
- Step 6: Modify the Collaboration Script 2-25
- Step 7: Copy the Input Data Files 2-26
- Step 8: Configure the e*Ways 2-27
- Step 9: Validate the Data 2-28
- Step 10: Analyze the Data 2-29

About the Validation Process

Overview

This section of the chapter provides background information you should know before customizing the validation schema for the initial load process. The level of validation performed in this phase is generally higher than would normally occur during production processing, and helps determine the quality of the data received. This phase separates the data file into two sets: one set of validated (or "good") data and one set of data that failed the validation ("bad" data).

Requirements

Before beginning the validation process, you need to obtain a flat file from the customer containing the data to be converted, as well as specific information about that data. Following is a complete list of the items that you should obtain.

- An ASCII flat file of the data that needs to be loaded into the database. The requirements for the data file to be loaded during the load phase are described under "Data File Requirements" beginning on page 3-14. The data extracts from the legacy systems should match these requirements as closely as possible, though there are procedures to validate the data and transform it into the required format.
- Any required valid value lists. Valid value lists are described under "Valid Value Lists" beginning on page 3-11.
- Any required exclusion lists. Exclusion lists are described under "Exclusion Lists" beginning on page 3-12.
- Information about the systems included in the records to be loaded into the data. You need to add this information to the database before loading member information.

About the Validation Phase

The first phase of the initial load process is the validation stage, and it transforms the input file into the required format (see "Data File Requirements" on page 3-14 for more information about the required format). When you run the data file through the validation e*Ways, three types of output files are created, one for data that passes the validations, one for data that does not pass the validations, and one describing the errors that occurred. The validation process consists of one schema with several e*Ways processing the data. You need to modify the e*Ways slightly before running the validation phase.

The validation occurs in two steps, the normalization step and the validation step. These steps are both defined in the collaboration rules file (**validate.tsc**).

Normalization

The processing rules for the normalization step of the validation phase are defined in the first section of **validate.tsc**. This section contains the "copy-strip" Monk functions that specify how the input structure is copied to the output structure. The default functions perform straight copies for each field, but you can modify these to perform checks and modify the values or formats if necessary.

Validation

The processing rules for validating the data that was copied in the normalization step are defined in the second section of **validate.tsc**. You can modify this section to include all of the required validations and to exclude any validations that are not required. For more information about the validation functions, see "Validation Functions" beginning on page 3-9.

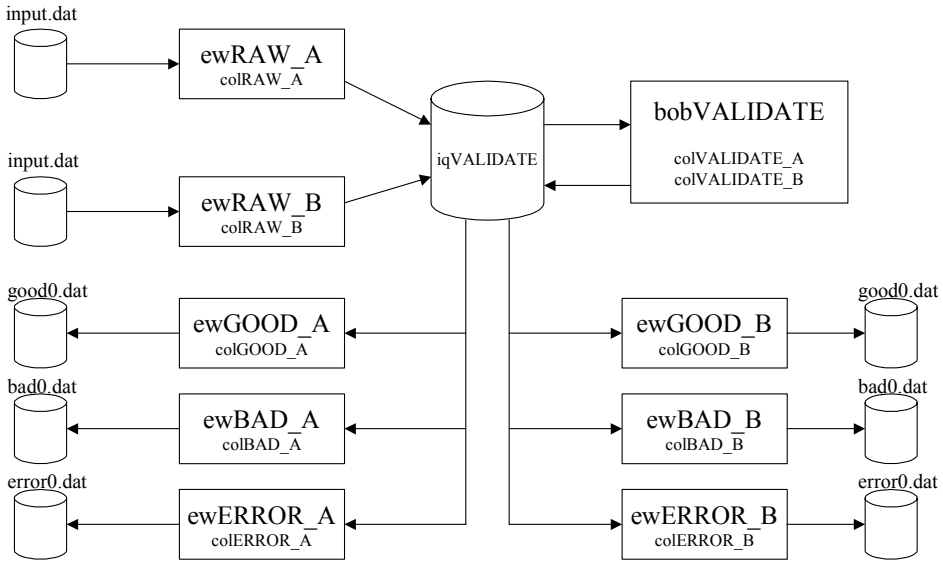
Validation Schema Architecture

This section provides an explanation of the components and files of the validation schema. Some of these components require modifications before running the validation.

The schema files are installed in the registry of the e*Gate server in **/<eGate>/Server/Registry/repository/ei_validate/runtime**. The schema also uses e*Index schema component files located in **.../repository/default**. The diagram displayed in "Step 10: Verify the Schema Directory Structure" in Chapter 2 of this guide illustrates the directory structure for the validation and load schemas.

The validation schema consists of four event types with their associated event type definitions (ETDs), one BOB, eight e*Ways (two sets of four for simultaneous processing), and the Intelligent Queue (IQ). This schema also includes a Monk library of **.monk** files used for defining the validations and check functions for the schema.

Phase 1 Schema Components



Event Types

The schema uses four event types that are associated with three customized ETDs and one generic ETD. These files help map the input file into three separate output files. One output file contains the data records that passed all validations, one contains data records that did not pass the validations, and one contains error information about the failed validations. The ETD files are installed in the e*Gate environment in **/<eGate>/Server/Registry/repository/ei_validate/runtime/monk_scripts/ui**.

- **etBAD**
  This event type is subscribed to by the **ewBAD_A** or **ewBAD_B** e*Way, depending on the original publisher of the **etRAW** event type. This event type defines the structure of the output file that contains the records that failed validation. **etBAD** is defined by the **etdGenericOutEvent.ssc** ETD.

- **etERROR**
  This event type is subscribed to by the **ewERROR_A** or **ewERROR_B** e*Way, depending on the original publisher of the **etRAW** event type. This event type defines the structure of the output file that contains an explanation of errors that occurred during validation. The records in the error output file correspond with the records in the bad data output file.

- **etGOOD**
  This event type is subscribed to by the **ewGOOD_A** or **ewGOOD_B** e*Way, depending on the original publisher of the **etRAW** event type. This event type describes the structure of the output file that contains records from the input file that passed all of the validations. It is defined by the **etdGOOD.ssc** ETD.

- **etRAW**
  This event type is subscribed to by the **ewRAW_A** or **ewRAW_B** e*Ways, and defines the structure of the input file. It is defined by the **etdRAW.ssc** ETD.

BOBs

The validation schema uses one BOB, **bobVALIDATE**, which subscribes to the **etRAW** event type. The event type is validated using **validate.tsc**, which publishes the **etGOOD** event type if all validation checks pass. It publishes the **etBAD** and **etERROR** pair of event types if one or more validation checks fail.

Note: *The collaboration rules file,* ***validate.tsc*** *is installed in* ***/<eGate>/Server/Registry/repository ei_validate/runtime/monk_scripts/ui****.*

---

e*Ways

The validation phase includes two identical sets of e*Ways, each set containing four different types of e*Ways. One e*Way in each set reads the input file, and three write to output files. Using two e*Ways of each type allows you to run a multi-threaded process. The configuration files for the e*Ways are located in **/<eGate>/Server/Registry/repository/ei_validate/runtime/configs/stcewfile**. The e*Ways include:

- **ewBAD_A** and **ewBAD_B**
  These e*Ways process the records that do not pass the validation process in the BOB. They subscribe to the **etBAD** event type, and write the error records to an output file. These e*Ways produce an output file of data that has been rejected. Each record in the output file contains a line number that corresponds to a line number in the error file.

- **ewERROR_A** and **ewERROR_B**
  These e*Ways process information about the records that do not pass the validation process. They subscribe to the **etERROR** event type, and write explanations of the errors encountered to an output file. The output file contains a line number and an error message. The error message explains why the corresponding record in the "bad" file was rejected.

- **ewGOOD_A.** and **ewGOOD_B**
  These e*Ways process the records that do pass the validation process in the BOB. They subscribe to the **etGOOD** event type, and write the good records to an output file that you can then use as the input file for the load phase.

- **ewRAW_A** and **ewRAW_B**
  These e*Ways process the records from the input data file, and subscribe to the **etRAW** event type. These e*Ways process the input records which are then picked up by the BOB. Each line of these files is published as the event type **etRAW**. You can feed multiple data files into each e*Way or you can use just one of these e*Ways to process the input data.

Monk Library

Three monk files are loaded when you install the validation schema. These files are located in the schema directory in **/runtime/monk_scripts/ui** and **/runtime/monk_library/ui**.

- **ui-check-functions.monk**
  This file defines the validation functions that you can call in the file **validate.tsc** in order to ensure that the fields in the output data file are formatted correctly. These functions are described in detail in "Validation Functions" on page 3-9.

- **ui-conv-lists.monk**
  This file defines the valid value lists and exclusion lists that you can use in the file **validate.tsc** to ensure that the fields in the output data file only contain allowed values. These lists are described in "Valid Value Lists" on page 3-11 and in "Exclusion Lists" on page 3-12.

- **ui-custom.monk**
  This file contains the commands you use to create demographic, alias, transaction, address, and telephone information lists, which are passed as parameters to the Monk APIs. You may need to customize this file using the e*Index Monk functions provided in **ui-fns.monk** (for more information about the functions in **ui-fns.monk**, see Chapter 5 of the *e*Index Global Identifier Technical Reference*).

Validation Schema Customizations

Most of the customizations made for each site are in the validation schema of the initial load. The components that you may need to modify include the following.

Event Type Definitions

The ETDs for the input and good output data (**etdRAW.ssc** and **etdGOOD.ssc**) are identical, apart from the name of the root node. You need

to modify **etdRAW.ssc** if it does not match the format of the flat file containing the legacy data (or the data file will need to be reformatted). SeeBeyond recommends that you only modify the **etdGOOD.ssc** ETD to match the ETD that is currently in use by the production **.dsc** scripts. This allows the same script to be used as in the production environment, ensuring that both production and load data are processed in the same manner.

If you modify the ETDs, you also need to modify the **validate.tsc** script to ensure that the data is copied correctly from the **etdRAW.ssc** ETD to the **etdGOOD.ssc** ETD. If you modify **etdGOOD.ssc**, you also need to modify the script **ui-load.dsc** (this file is for the load phase of the process).

BOBs

The schema uses the same collaboration script, **validate.tsc**, for each collaboration in **bobVALIDATE**. Modifications to the collaboration script affect all validation feeds. You can copy this file, make modifications to the new file, and associate it with the appropriate collaborations if you need different sets of processing rules for your input data.

The file **validate.tsc** defines the rules for how to copy data from the input file to the output file, and specifies which validations to perform. This file performs two tasks: 1) copy the input data and 2) validate the copied data. Generic normalizations, such as defining default data for empty fields, should be defined in the copy section of the script. After running a test validation, you may find common errors that require generic normalization. For example, if a specific field must be left-justified, but is commonly found not to be, you can specify that it be left-justified, eliminating the error and allowing the record to be processed. The validation portion of the script contains specific validation checks for each field (for more information about validation checks, see "Validation Functions" beginning on page 3-9.

Note: *You may want to review the normalizations that are required here and perform the same normalizations during the production process, as several of the same data issues are likely to occur in the production environment.*

---

e*Ways

You can create additional e*Ways to load input data files, based on the existing e*Ways **ewRAW_A** and **ewRAW_B**. The number of e*Ways should be based on the number of input files to process, although you can process multiple files through each e*Way. If you do add e*Ways to the schema, you need to add a collaboration for each e*Way to the **bobVALIDATE** BOB in order to process the data published by the new e*Ways. For each collaboration you add to the BOB, you also need to add three e*Ways – one to subscribe to each of the three output event types (**etGOOD**, **etBAD**, and **etERROR**) from the collaboration. For example, creating a new e*Way (**ewRAW_C**) requires a new collaboration (**colVALIDATE_C**) and three new

e*Ways (**ewGOOD_C**, **ewBAD_C**, and **ewERROR_C**). You can base all of these on the existing schema components.

If any file-based processing is required (such as modifying specific field values so they match the table codes defined for e*Index), you can add collaborations to the e*Ways processing the input data to make the required modifications.

Monk Scripts

In order to define and specify valid value and exclusion lists, you need to modify the Monk files. Valid value lists are described on page 3-11 and exclusion lists are described on page 3-12. These lists are defined in **ui-conv-lists.monk**, and should be customized for each customer site. You can also customize the check functions defined in **ui-check-functions.monk**. These functions are described beginning on page 3-9.

You may also want to modify the Monk lists defined in **ui-custom.monk** to make sure the fields in each list are formatted as required. See chapter 5 of the *e*Index Global Identifier Technical Reference* for more information.

Validation Functions

The **ui-check-functions.monk** file includes several validation functions that you can use to ensure that the data being loaded into the database is in the proper format. These functions are called within the collaboration file **validate.tsc**, which copies data from the input structure to the output structure and verifies the data in the output file. Call these functions in the second section of **validate.tsc**, "Validation checks". This section of the file contains default validations that you can customize as needed.

Tip: *To disable one of the existing validations without deleting it, change* `the #t in the IF statement to #f.`

In the parameters for the functions listed below, the **input_path** is the location of the field in the file structure and **field_name** is the name of the field being validated. The validation functions include:

- **CheckNotNull**
  This function checks the specified data elements for null values and returns an error string if a null value is found. The syntax for this function is:

```
(CheckNotNull input_path field_name)
```

- **CheckLeftJust**
  This functions checks whether a field is left justified. If the specified field is not left justified, the function returns an error string. The syntax for this function is:

```
(CheckLeftJust input_path field_name)
```

- **CheckRightJust**
  This functions checks whether a field is right justified. If the specified field is not right justified, the function returns an error string. The syntax for this function is:

```
(CheckRightJust input_path field_name)
```

- **CheckChars**
  This functions checks whether a field contains only the types of characters that you specify. The syntax for this function is:

```
(CheckChars input_path field_name flags)
```

  The **flags** parameter is three characters long, and defines the types of characters that are valid for the field. The flags can be entered as described below:

  - The first character in the flag can be either an **A** or a hyphen (-). An **A** indicates that alphabetic characters are allowed; a hyphen indicates that alphabetic characters are not allowed.
  - The second character can be either an **N** or a hyphen. An **N** in this field indicates that numeric characters are allowed; a hyphen indicates that numeric characters are not allowed.
  - The third character can be either a **P** or a hyphen. A **P** indicates that punctuation characters are allowed; a hyphen indicates that punctuation characters are not allowed.

  For example, the flag "**A-P**" means that alphabetic and punctuation characters are allowed, but numeric characters are not. The flag "**-N-**" means that only numeric characters are allowed.

- **CheckValidValues**
  This function checks whether the specified field contains a value that is included in the specified valid value list. If the value is not in the list, then an error is returned. The syntax is as follows:

```
(CheckValidValues input_path field_name vvlist)
```

  where **vvlist** is the name of the valid values list containing the values that are allowed for the specified field.

- **CheckExclValues**
  This function checks whether the specified field contains a value that is included in the specified exclusion list. If the value is in the list, then an error is returned. The syntax is as follows:

```
(CheckExclValues input_path field_name exclist)
```

  where **exclist** is the name of the exclusion list containing the values that are not allowed for the specified field.

- **CheckFormat**
  This function checks whether the specified field is in the correct format. If

the field is not in the specified format, then an error is returned. The syntax is as follows:

```
(CheckExclValues input_path field_name format)
```

The possible values you can enter for the format are as follows:

- **UI_DATE_FORMAT**
  This format indicator changes the date format to **YYYY-MM-DD**.
- **UI_TIME_FORMAT**
  This format indicator changes the time format to **HH:MM:SS**.
- **UI_SSN_FORMAT**
  This format indicator changes the social security number format to **NNN-NNN-NNNN**.

Tip: *You can modify the format for these values in the last section of* ***ui-conv-lists.monk*** *by changing the values for these variables:* ***define UI-DATE-FORMAT****,* ***define UI-TIME-FORMAT****, and* ***define UI-SSN-FORMAT****. The formats are defined using standard Monk.*

Valid Value Lists

If a valid value list is specified for a data field, then the value in that field must be one of the values in the associated valid value list. Typically, the values you specify are the codes associated with the actual entity (for example, for marital status, you might specify "S" to indicate "single", "M" to indicate "married", and so on). If the corresponding field can be null, then the null value is implicitly one of the values in the valid value list for that field. An empty valid value list implies that no value found in the corresponding field is permitted. Valid values may not start or end with a blank character, but they may contain one or more blank characters in the middle. The customer must provide a valid value list for each field for which valid values are required. If a value list is called in **validate.tsc** but one is not provided, then an empty list is assumed.

The following valid value lists are defined in **ui-conv-lists.monk**. You can modify these lists and define new lists if necessary. Remember that the fields in e*Index are configurable, so the actual field names displayed on the GUI may differ from the field names below.

- **functionVVL**: The valid value list for functions (also known as events or transaction types), such as adds, merges, and so on.
- **systemVVL**: The valid value list for the system field.
- **deathVVL**: The valid value list for the death field.
- **sexVVL**: The valid value list for the sex field.
- **religionVVL**: The valid value list for the religion field.
- **raceVVL**: The valid value list for the race field.
- **ethnicityVVL**: The valid value list for the ethnicity field.

- **languageVVL**: The valid value list for the language field.
- **countryVVL**: The valid value list for the country field.
- **suffixVVL**: The valid value list for the suffix field.
- **marital_statusVVL**: The valid value list for the marital status field.
- **vipVVL**: The valid value list for the VIP field.
- **veteranVVL**: The valid value list for the veteran status field.
- **vet_mil_statusVVL**: This valid value list is obsolete and can be disregarded.
- **person_categVVL**: The valid value list for the person category field.
- **dorVVL**: The valid value list for the DOR (district of residence) field.
- **deptVVL**: The valid value list of departments.
- **sourceVVL**: The valid value list of sources.
- **non_unique_typeVVL**: The valid value list of non-unique ID types (also known as other ID types).
- **address_typeVVL**: The valid value list for the address type field.
- **phone_typeVVL**: The valid value list for the telephone type field.
- **military_statusVVL**: The valid value list for the military status field.

Exclusion Lists

If an exclusion list is specified for a data field, then the value in that field cannot be one of the values in the associated exclusion list. If an exclusion list value is found, it is converted to null. No value in an exclusion list may be null, but the list itself may be empty. An empty exclusion list implies that any value found in the corresponding field is permitted. Exclusion values may not start or end with a blank character, but they may contain one or more blank characters in the middle. You can use exclusion lists to ensure that invalid social security numbers (999-99-9999), invalid telephone numbers (999-999-9999), etc. are not allowed. If an exclusion list is required but not provided, then an empty list is assumed.

The following exclusion lists are defined in **ui-conv-lists.monk**. You can modify these lists and define new lists if necessary. Remember that the field names are configurable, so the fields on the GUI may not match the field names below.

- **zipEXL**: This is the exclusion list for the zip code field.
- **ssnEXL**: This is the exclusion list for the social security number field.
- **dobEXL**: This is the exclusion list for the date of birth field.

Important Validation Information!

While the default validations included in **validate.tsc** are comprehensive, there may be certain custom validations that you need to add to the file. In addition to your customizations, SeeBeyond recommends that you add validations to check the following:

- Invalid day or month in a date field
- Missing system in the local ID field
- Missing non-unique ID or non-unique ID type
- Missing first or last name in the alias field

It is important to note that when the following errors occur, the record will pass the default validation, but data will be removed. You may also want to add validations for these scenarios.

- Missing or invalid identifier in the Auxiliary segment (all data in the segment is removed)
- Missing first non-unique ID, alias, address, or telephone (all subsequent entries are removed from these repeating fields if the first instance is empty)

Output Files

Validating the data file produces three types of output files: **good data**, **bad data**, and **error data**. You can define the names of these files when you modify the ei_validate schema. The good data file contains all of the records that conform to format requirements; the bad data file contains records with errors; and the error data file describes each record that was written to the bad data file. After you process the input file, you can fix any errors in the input data file by checking the error data and bad data files. Once you fix the errors, you can reprocess the input data. Information from these files can also tell you if you need to modify the current setup of the field validations.

By default, the output files are placed in the e*Gate client directory in **/<eGate>/client/eIndex/validate/output/a** or **/<eGate>/client/eIndex/validate/output/b**.

Data File Requirements

Overview

This section describes the format of the data file to be loaded into the database. If records are not in the format described, you need to modify either the data file format or the input ETD. The data file you obtain from the legacy systems generally require the validations and modifications performed in the validation schema before it can be loaded into the database.

Formatting Guidelines

In order for you to load legacy data into the e*Index database, the input file received from the customer must be reformatted as follows:

- Each record consists of two types of information: Transaction details and identification details. These are delimited by a pair of angled brackets (<>).
- The records must be delimited. Each segment is separated by an ampersand (&), each field is separated by a pipe (|), and each sub-field is separated by a caret (^). When a field can repeat, each repetition is separated by a tilde (~). The segments appear as follows:

  EVNT segment <> ID segment & DEMO Segment & AUX segment <>
- Each record must be contained on a single line.
- Each record may contain only ASCII printable characters, and alphabetic characters must be upper case.

  For information about each field, see the Input ETD Structure table below.

Note: *This record format corresponds to the format of records passed to the production collaboration script for e*Index. This should be reviewed for each site to simplify where applicable. For example, fields for which the sending systems do not collect data can be removed.*

Table 3-1 – Input ETD Structure

Transaction Details

Field	Description	Repeating?	Required?
segment_id	"EVNT"	No	Yes
msg_id	This field is not used.	No	No
event_type_code	Always leave this field blank. The system will automatically determine the transaction type.	No	No

Field	Description	Repeating?	Required?
user_id	The user ID of the user who performed the transaction.	No	No
assigning_system	The system code for the system on which the transaction was performed.	No	No
source	The source code of the application on which the transaction was performed.	No	No
department	The department code for the transaction.	No	No
terminal_id	The ID of the terminal on which the transaction was performed.	No	No
date_of_event	The date the transaction occurred. If this is left blank, the record will be rejected. If supplied, the date should be in the format **YYYY-MM-DD**. This field can be reformatted during the validation process. For more information, see "Important Information About Dates and Times of Events" later in this chapter.	No	Yes
time_of_event	The time the transaction occurred. If this is left blank, the record will be rejected. If supplied, it should be in the format **HH:MM:SS** using a 24-hour clock (for example, 23:59:59). This field can be reformatted during the validation process. For more information, see "Important Information About Dates and Times of Events" later in this chapter.	No	Yes

Identification Details

Field	Description	Repeating?	Required?
segment_id	"ID"	No	Yes
eid_1	Always leave this field blank.	No	No
local_id	The person's local identifier at a specified system. This field has two sub-fields: ID and system. For example, if the local ID **12345** was assigned within the system **SeeBeyond**, which has a code of **SBYN**, this field should appear like this: `|12345^SBYN|` ***Note:*** *If multiple local IDs are specified, they will all be linked to the same EID.*	Yes	Yes

Field	Description	Repeating?	Required?
non_unique_id	The person's non-unique identifiers. This field has two sub-fields: ID and type. For example, if a person's account number is **003487**, and the type code for account is **ACCT**, this field should appear like this: |003487^ACCT|	Yes	No
segment_id	"DEMO"	No	Yes
person_category	The code for the person category to which the person is assigned.	No	Yes
person_name	The name of the person. This field consists of five sub-fields. **last_name:** The person's last name. **first_name**: The person's first name. **middle_name:** The person's middle name. **title:** The title code of the person's title. **suffix:** The suffix code of the person's suffix to their name.	No Yes Yes No No No	
person_alias	The alias names for the person. This consists of three sub-fields: **last_name:** The alias last name. **first_name**: The alias first name. **middle_name:** The alias middle name.	Yes	No
alt_name	Alternative names associated with this person. This field consists of five sub-fields: **maiden_name:** The person's maiden name. **spouse_name**: The name of the person's spouse. **mother_name:** The name of the person's mother. **fathers_name:** The name of the person's father. **mothers_maiden_name:** The maiden name of the person's mother.	No	No

Field	Description	Repeating?	Required?
date_of_birth	The person's date of birth, in YYYY-MM-DD format. ***Note:*** *The format can be modified into the correct format during the validation process.*	No	Yes
time_of_birth	The time the person was born, in HH:MM:SS format on a 24-hour clock. ***Note:*** *The format can be modified into the correct format during the validation process.*	No	No
sex	The table code of the person's gender.	No	Yes
marital_status	The table code of the person's marital status.	No	No
SSN_number	The person's social security number in ###-##-#### format. ***Note:*** *The format can be modified into the correct format during the validation process.*	No	No
drivers_license	The drivers license details for the person. This has two sub-fields: **state_country:** The state or country that issued the drivers license. **number:** The drivers license number.	No	No
race	The table code of the person's race.	No	No
ethnic_group	The table code of the person's ethnic group.	No	No
nationality	The table code of the person's nationality.	No	No
religion	The table code of the person's religion.	No	No
language	The table code of the language spoken by the person.	No	No
death	Death information about the person. This field consists of three sub-fields: **flag:** An indicator of whether the person is deceased. Should be **Y** if deceased. **date_of_death:** If deceased, the date of death. **death_certificate_number:** The ID number on the death certificate.	No	No

Field	Description	Repeating?	Required?
birth_place	The location in which the person was born. This field consists of three sub-fields: **city:** The city where the person was born. **state:** The state where the person was born. **country:** The country code where the person was born.	No	No
vip	The table code of the person's VIP status.	No	No
veteran_status	The table code of the person's veteran status.	No	No
military	The military details for the person. This field consists of three sub-fields: **status:** The code of the person's military status. **rank_grade:** The person's military rank or grade. **branch:** The military branch in which the person has served.	No	No
citizenship	Citizenship for the person	No	No
pension	The pension details for the person. This field consists of two sub-fields: **number:** The person's pension card number. **expiration_date:** The expiration date of the pension card.	No	No
repatriation_number	The person's repatriation number.	No	No
district_of_residence	The code of the district of residence in which the person resides.	No	No
LGA_code	The LGA code for the person.	No	No
address	Address information for the person. This field consists of eleven of sub-fields: **type:** The table code for the type of address. **street_1:** The first line of the street address. **street_2:** The second line of the street address. **street_3:** The third line of the street address.	Yes maximum of three addresses	No

Field	Description	Repeating?	Required?
	street_4: The fourth line of the street address. **city:** The city or suburb of the address. **state_or_province:** State or province **zip:** The zip code of the address. **zip_ext:** The zip code extension of the address. **county:** The county where the address is located. **country:** The table code of the address's country. ***Note:*** *If address information is included in an Event, the following fields must be present for each address:* ***type***, ***street_1***, *and* ***city***.		
phone	Telephone information for the person. This field consists of three sub-fields: **type:** The table code of the telephone type. **phone_number:** The telephone number. **phone_ext:** The extension to the telephone number. ***Note:*** *If telephone information is included in an Event, the* ***type*** *and* ***phone_number*** *fields must be present for each telephone number.*	Yes maximum of three telephone numbers	No
segment_id	"AUX"	No	Yes
class	Five 20-character strings for site-specific purposes.	Yes maximum of five	No
string	Additional strings for site-specific purposes. The first six are a maximum of 40 characters. Strings seven to nine are a maximum of 100 characters. The tenth string is a maximum of 255 characters	Yes maximum of ten	No
date	Five miscellaneous date fields. These must be **YYYY-MM-DD** format. ***Note:*** *The format can be modified into the correct format during the validation process.*	Yes maximum of five	No

Sample Inbound ETD

Below is a sample data record that follows the format described above.

```
EVNT||||SBYN|SBYN|||2000-04-27|12:00:00<>ID||0000000012^SBYN|&
DEMO||BLACKBIRD^JULIE^^^||^^^^|1992-04-12||F||471-05-1547|^|B||
|||^^|^^|||^^||^||||H^712 SOUTH WINSTON
DR^^^^PASADENA^CA^91101^^^US|<>
```

Important Information About Dates and Times of Events

The initial load process uses the date_of_event and time_of_event fields to determine which data is the most current. This date determines which records are updated and which records are inserted as new records. It is essential that these fields be populated and in the correct format. Records that are missing either the date or time, or whose date or time is incorrectly formatted, are rejected. It is very important that these fields be accurate since the system determines which information to retain based on the date and time of event.

When processing records, if the date and time of event of the incoming record is later than or equal to the create/update date of a matching record in the database, the existing record is updated with the new information. If the date and time of event of the incoming record is earlier than the create/update date of a matching record in the database, the incoming record is regarded as old. Local ID and system information from the incoming record is updated in the existing record, and the remaining information is discarded. The date and time of event is stored in the update date field for any records that are updated or inserted.

It is understandable that not all systems will record the time of each event. In such cases, you should determine a default time to populate into this field. If you have data records that do not have event dates, you will need to determine a default for these fields as well. Typically, records without event dates are from older systems, and are most likely older records. SeeBeyond recommends using a default date well into the past (such as 01/01/1950) to ensure that no new information is overwritten.

Standardizing the Data File

Overview

Before loading the data file into the e*Index database, you need to standardize the records in the file. You must complete the following steps to standardize the data:

- Step 1: Obtain the Data File and Lists
- Step 2: Modify **ui-conv-lists.monk**
- Step 3: Modify **ui-check-functions.monk**
- Step 4: Modify **ui-custom.monk**
- Step 5: Modify the ETDs
- Step 6: Modify the Collaboration Script
- Step 7: Copy the Input Data Files
- Step 8: Configure the e*Ways
- Step 9: Validate the Data
- Step 10: Analyze the Data

Note: *If you need to create additional e*Ways and collaborations to process the data files, make sure they are completed before you begin the following steps.*

Step 1: Obtain the Data File and Lists

You need to obtain an ASCII flat file containing the data to be loaded into the e*Index database. This flat file should match as closely as possible the format described earlier in this guide under "Data File Requirements" beginning on page 3-14. You can work with your SeeBeyond representative to create valid value lists and exclusion lists for the fields for which they are required. These lists are described beginning on page 3-11.

Step 2: Modify ui-conv-lists.monk

Once you have developed lists of fields that require valid value or exclusion lists, and determined the values to assign to each list, you can define the lists in **ui-conv-lists.monk**.

To modify ui-conv-lists.monk

Before you begin:

- ✓ Complete "Step 1: Obtain the Data File and Lists"
- ✓ Make sure you have the valid value and exclusion lists
- ✓ Make sure the initial load program is installed as described in chapter 2 of this guide

1. In the e*Gate Enterprise Manager or any text editor, open the file **ui-conv-lists.monk** (located in **/<eGate>/server/registry/repository/ei_validate/runtime/monk_scripts/ui**).

2. In the first section of the file, define any necessary exclusion lists. The syntax for these lists is:

```
(define <listname> #("<value>" "<value>"))
```

where <listname> is the name of the exclusion list (usually the field name followed by "EXL") and <value> is any value that is not allowed in the specified field. For example, the following exclusion list for zip codes indicates that the values "00000" and "11111" are not allowed in the zip code field.

```
(define zipEXL   #("00000" "11111"))
```

3. In the second section of the file, define any necessary valid value lists. The syntax for these lists is:

```
(define <listname> #("<value>" "<value>")
```

where <listname> is the name of the valid value list (usually the field name followed by "VVL") and <value> is any value that is allowed in the specified field. For example, the following valid value list for genders indicates that the values "M", "F", and "U" are the only values allowed in the gender field.

```
(define sexVVL  #("M" "F" "U"))
```

Tip: *You can indicate null values for both exclusion and valid value lists by adding empty quotes ("") to the list.*

4. The last section of this file defines the date, time, and social security number formats that are called in the CheckFormat function (defined in **ui-check-functions.monk**). You can change the defined formats if necessary. For more information, see the tip on page 3-11.

5. Save and close the **ui-conv-lists.monk** file.

Note: *If you edited this file from the e*Gate Enterprise Manager, remember to commit the file to the sandbox, and then promote it to runtime. You can also commit the file using the* ***stcregutil*** *command.*

6 Continue to "Step 3: Modify **ui-check-functions.monk**."

Step 3: Modify ui-check-functions.monk

The **ui-check-functions.monk** file contains several functions that you can call in the collaboration rules file to validate the fields in the processed data. You should not need to modify this file, but you can create additional check functions if required. These functions can easily be modified using the Monk scripting language.

▶ To modify ui-check-funcs.monk

Before you begin:

- ✓ Complete "Step 2: Modify **ui-conv-lists.monk**"
- ✓ Obtain information about requirements for additional check functions or changes to existing check functions

1 Navigate to **/<eGate>/server/registry/repository/ei_validate/runtime/monk_scripts/ui** and open the file **ui-check-functions.monk** in any text editor.

2 Modify any existing check function, or define new check functions.

Caution! *You should be very familiar with the Monk scripting language and the format of the incoming data file before modifying this file.*

---

3 Save and close the **ui-check-functions.monk** file.

Note: *If you edited this file from the e*Gate Enterprise Manager, remember to commit the file to the sandbox, and then promote it to runtime. You can also commit the file using the* ***stcregutil*** *command.*

---

4 Continue to "Step 4: Modify **ui-custom.monk**."

Step 4: Modify ui-custom.monk

If necessary, you can modify **ui-custom.monk** to reformat some of the fields defined in the Monk lists. For more information about the functions you can use in this file, see chapter 5 of the *e*Index Global Identifier Technical Reference*.

▶ To modify ui-custom.monk

Before you begin:

- ✓ Complete "Step 3: Modify **ui-check-functions.monk**"

✓ Review the information in chapter 5 of the *e*Index Global Identifier Technical Reference*

1 Navigate to **/<eGate>/server/registry/repository/ei_validate/runtime/monk_library/ui** and open the file **ui-custom.monk** in any text editor.

2 Use the functions included in **ui-fns.monk** to modify the fields included in each Monk list.

3 Save and close **ui-custom.monk**.

Note: *If you edited this file from the e*Gate Enterprise Manager, remember to commit the file to the sandbox, and then promote it to runtime. You can also commit the file using the* ***stcregutil*** *command.*

---

4 Continue to "Step 5: Modify the ETDs."

Step 5: Modify the ETDs

If the structure of the data extract file that you are validating does not match the current structure of the predefined inbound ETD (**etdRAW.ssc**), you must modify either the data file or the ETD so the structures match. In addition, SeeBeyond recommends that you modify the output ETD **etdGOOD.ssc** to match the ETD being used in production by the collaboration script (**uidb.dsc**). You do not need to modify **etdERROR.ssc**.

Important! *Remember that if you modify either of these files, you'll need to modify* ***validate.tsc*** *to match the changes.*

▶ To modify the ETDs

Before you begin:

✓ Complete "Step 4: Modify **ui-custom.monk**"

1 In the e*Gate Enterprise Manager or any text editor, open the appropriate ETD file (located in **/<eGate>/server/registry/repository/ei_validate/runtime/monk_scripts/ui**).

2 Make the necessary modifications to the defined data structure.

Caution! *You should be very familiar with the Monk scripting language and the format of the incoming data file before modifying this file.*

---

3 Save and close the ETD file.

Note: *If you edited this file from the e*Gate Enterprise Manager, remember to commit the file to the sandbox, and then promote it to runtime. You can also commit the file using the* ***stcregutil*** *command.*

---

4 Continue to "Step 6: Modify the Collaboration Script."

Step 6: Modify the Collaboration Script

The collaboration script file is where you specify how fields are copied from the input structure to the output structure. This file must reference the validation and exclusion lists you defined, and uses the check functions to validate the copied files. You should modify this file to include all of the validations you require. In addition, if you modified either ETD, you need to change this file to match.

▶ To modify the collaboration script

Before you begin:

- ✓ Complete "Step 5: Modify the ETDs"
- ✓ Obtain information about required data validations and normalizations for each field

1 In the e*Gate Enterprise Manager or any text editor, open **validate.tsc** (located in **/<eGate>/server/registry/repository/ei_validate/runtime/monk_scripts/ui**).

Caution! *You should be very familiar with the Monk scripting language and the format of the incoming data file before modifying this file.*

---

2 Currently, the **max_err_cnt** is set to **5**. This means that only the first five errors for each record are reported. You can modify this value as required.

3 In the "Normalise Data" section, make any necessary modifications to the copy statements to ensure that the fields are copied correctly. You can add formatting rules to reformat the fields (such as those defined in **ui-fns.monk**).

4 In the "Validate the Normalised Data" section, modify the check functions or add new check functions as required.

To add a new check function, you can copy an existing IF statement and modify the details.

To modify an IF statement, you only need to modify the test components. The body of each IF statement is identical.

To disable a test, simply change the **#t** to **#f**. For example, the first sample below checks for a null value in date_of_event. The second sample does not check for left justification in that field.

```
...(and (< cur_err_cnt max_err_cnt) #t (not-empty-string? (set!
work (CheckNotNull ~output%etdGOOD.EVNT.EVN.date_of_event
"date_of_event"))))
...
```

```
...(and (< cur_err_cnt max_err_cnt) #f (not-empty-string? (set!
work (CheckLeftJust ~output%etdGOOD.EVNT.EVN.date_of_event
"date_of_event"))))
...
```

5 Save and close the collaboration script file.

Note: *If you edited this file from the e*Gate Enterprise Manager, remember to commit the file to the sandbox, and then promote it to runtime. You can also commit the file using the* ***stcregutil*** *command.*

---

6 Continue to "Step 7: Copy the Input Data Files."

Step 7: Copy the Input Data Files

When you installed the initial load schemas, you created a file structure in **/<eGate>/client** for the initial load data files. Use these directories to store your input and output data files. By default, the e*Ways are configured to read and write files in these directories. You can create a separate directory structure for the data files if you want, but remember to configure each e*Way with the new pathnames.

▶ To copy the input data files

Before you begin:

- ✓ Complete "Step 6: Modify the Collaboration Script"
- ✓ Determine the number of e*Ways you'll be using, and which input data files will be processed through each e*Way

1 If you created additional e*Ways, you need to create directories for the input and output files.

To create directories for the input files, navigate to **/<eGate/client/eIndex/validate/input**, and create additional folders for each new **ewRAW_*** e*Way. For example, if you created a new e*Way named **ewRAW_C**, you should create a folder in this directory named "**c**".

To create directories for the output files, navigate to **/<eGate/client/eIndex/validate/output**, and create additional folders for each new **ewRAW_*** e*Way. For example, if you created a new e*Way named **ewRAW_C**, you should create a folder in this directory named "**c**".

2 Copy the input data files to be processed into the appropriate directories in **/<eGate/client/eIndex/validate/input**. For example, copy the files to be processed by **ewRAW_A** in the "**a**" directory, by **ewRAW_B** in the "**b**" directory, and so on.

3 To simplify e*Way configuration, modify the name of each input file so they have a **.dat** extension (for example, input.dat).

4 Continue to "Step 8: Configure the e*Ways."

Step 8: Configure the e*Ways

You may need to make some minor modifications to the e*Ways, depending on how you want to use the performance testing options, where you placed the input data file, and where you'd like the output files to be written.

Note: *By default, all of the "_A" e*Ways are configured to automatically start, and the "_B" e*Ways are not. If you want all of your e*Ways to start automatically when you open the e*Gate Monitor, you should configure them to do so.*

▶ To configure the e*Ways

Before you begin:

✓ Complete "Step 7: Copy the Input Data Files"

1 To configure the e*Ways that read the input files, do the following for each **ewRAW_*** e*Way.

- In the e*Gate Enterprise Manager open the Properties window for the e*Way.
- Modify the Startup, Advanced, and Security options as needed.
- Modify the Configuration file as needed.
  - You do not need to modify the General Settings or the Outbound (send) settings, but you can modify the Performance Testing parameters as needed.
  - If you did not copy the input data files into the default directories, or you did not give them a **.dat** extension, modify the Poller (inbound) settings by entering the new pathname in the **PollDirectory** parameter and/or entering the new file extension in the **InputFileMask** parameter. Modify the remaining parameters as needed.

2 To configure the e*Ways that write the output data files, do the following for each **ewGOOD_***, **ewBAD_***, and **ewERROR_*** e*Way.

- In the e*Gate Enterprise Manager open the Properties window for the e*Way.
- Modify the Startup, Advanced, and Security options as needed.
- Modify the Configuration file as needed.
  - You do not need to modify the General Settings or the Poller (inbound) settings, but you can modify the Performance Testing parameters as needed.
  - If the path in which you want to store the output files is not the default path in **/<eGate/client**, modify the Outbound (send) settings by entering the new pathname in the **OutputDirectory** parameter. You can also modify the file name for the **OutputFileName** parameter. Modify the remaining parameters as needed.

3 Continue to "Step 9: Validate the Data."

Step 9: Validate the Data

Once you have made the necessary modifications, you are ready to run the validation schema to validate the input data.

▶ To validate the data

Before you begin:

- ✓ Complete "Step 8: Modify the e*Ways"
- ✓ Make sure the input files are in the appropriate directories with the correct file names (as defined in the e*Way configuration file).

1 Start the control broker for the ei_validate schema using the **stccb** command.

2 Start the ei_validation schema in the e*Gate Monitor. If you configured all of the e*Ways you are using to start automatically, they will start on their own. Otherwise, you need to start them manually.

3 As the input files are processed, the output files are placed in **/<eGate>/client/eIndex/validation/output/*** (if you used the default configuration).

4 While the process is running, continue to "Step 10: Analyze the Data Output."

Step 10: Analyze the Data Output

While running the validation schema, view the output files to check for any errors. The **bad*.txt** file contains a list of all records that did not pass the validation tests. For more information about the errors that occurred, view the **error*.txt** file. The entries in these files are numbered, and each line number in **bad*.txt** corresponds with a line number in **error*.txt**. View the files together to see what types of errors occurred.

If you noticed a pattern in the errors, you may want to stop the conversion and correct the error either in the input file or by modifying the validations. For example, if several records are failing the validation because a specific value is not included a valid value list, you may want to modify the valid value list to allow that value. Or, if several records are failing because specific fields are not left justified, you may want to specify in the "copy" command for that field that they be made left-justified. In addition, you can use the error information from **bad*.txt** and **error*.txt** to fix records in the input data file.

Once you have made the necessary modifications, repeat steps 9 and 10 until you feel you have defined all the required validations and made any necessary changes to the input data, the valid value lists, and the exclusion lists.

Chapter 4

Loading the Validated Data

About this Chapter

Overview

This chapter presents the background information and the step-by-step instructions you need to load legacy data into the e*Index Global Identifier database.

The following diagram illustrates the contents of each major topic in this chapter. For the page numbers on which specific topics appear, see the next page of this chapter.

About the Load Schema — Learn about the architecture of the load schema, and how data is processed

Perform the Data Load — Learn the steps required to load data into the database using the load schema

What's Inside

This chapter provides background information and instructions related to the topics listed below.

Learning About the Load Schema 2-3
Loading the Data File 2-9
- Step 1: Copy the Data Files 2-9
- Step 2: Configure the Database 2-10
- Step 3: Modify the Monk Files 2-11
- Step 4: Modify the ETDs 2-11
- Step 5: Modify the Collaboration Script 2-12
- Step 6: Configure the e*Ways 2-12
- Step 7: Stop the Archive Log 2-14
- Step 8: Load the Data 2-14
- Step 9: Fix any Records in the Error File 2-15
- Step 10: Restart the Archive Log 2-15
- Step 11: Run Database Reports 2-16

Learning About the Load Schema

Overview

This section of the chapter provides background information you should know before loading the data from the validation phase into your e*Index database. The load phase writes good data records directly to the database, and produces an output file containing any records that could not be loaded into the database.

Requirements

Before beginning the load process, make sure that all of the input files containing legacy data to be loaded have been validated. The "good" output files from the validation phase are the new input files for the load phase. Before running the load schema, you need to enter system information and codes into the e*Index database. Make sure you have all of the system names and their associated codes before beginning the load process.

About the Load Phase

Once you complete the validation process, the data is ready to be loaded into the e*Index database. During the load process, you can monitor the progress of the data load by reviewing the log files. Depending on the size of the database and the number of records to process, this may take several days. Any records that cannot be processed due to errors are written to an error file. After completing the first data load, you can fix the records written to this file, and then rerun the schema against the error file to ensure that you process all records in the data file. You should also run reports against the database to check for any potential duplicates you may need to resolve.

The load process consists of one schema with several e*Ways processing the data. This schema uses a subscriber pool to load data into the database more quickly and efficiently. You may need to modify the e*Ways slightly before running the validation phase.

Output Files

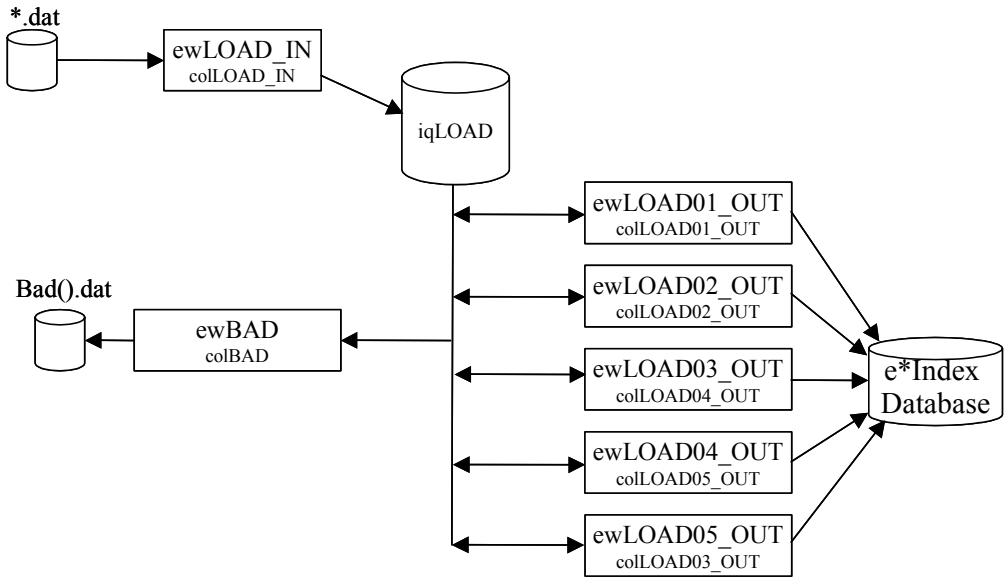
When you process data through the load phase, one output file is produced, and by default is written to **/<eGate>/client/eIndex/load/output/a** (or **/b**). You can change the output directory by modifying the **ewBAD** e*Way. The output file contains information about any records that were rejected during the load process. You can fix the error records in this output file and then reprocess those records through the load schema. No matter how many input files you use for the load schema, only one output file is created to store

the bad data. Subsequent runs of the schema will append the new bad data to the end of the existing output file.

Load Schema Architecture

This section provides an explanation of the components and files of the load schema. You may need to modify some of these components before loading the data. The schema files are installed in the registry of the e*Gate server in **/<eGate>/Server/Registry/repository/ei_load/runtime**. The schema also uses component files from the e*Index schema. These files are located in **/<eGate>/Server/Registry/repository/default**. The diagram under "Step 10: Verify the Schema Directory Structure" in chapter 2 illustrates the directory structure for the validation and load schemas.

Phase 2 Schema Components



The load schema consists of two event types with their associated event type definitions (ETDs), seven e*Ways (using subscriber pooling), and the IQ. This schema also includes a Monk library of **.monk** files that define the processing rules for processing records, define Monk lists for the APIs, and define database connectivity functions.

Event Types

The schema includes two event types that are associated with one customized ETD and one generic ETD. These files are used to map the input file into the e*Index database and to one output file that lists all records that

contained errors. The ETD files are installed in the e*Gate environment in **/<eGate>/Server/Registry/repository/ei_load/runtime/monk_scripts/ui**.

- **etBAD**
  This event type is subscribed to by the **ewBAD** e*Way, and is published to by the **ewLOAD##_OUT** e*Ways. This event type defines the structure of the output file that contains the records that could not be loaded into the database. **etBAD** is defined by the **etdGenericOutEvent.ssc** ETD.
- **etLOAD**
  This event type is subscribed to by the **ewLOAD_IN** and **ewLOAD##_OUT** e*Ways. Both e*Ways publish to **etLOAD** as well. This event type describes the structure of the input file and of the records that are inserted into the database. It is defined by the **etdGOOD.ssc** ETD in the load schema, which should be identical to the **etdGOOD.ssc** file used in the validation schema.

Collaboration Rules

One collaboration rules file is included in the load e*Way. This file is **ui-load.dsc**, and is installed in **/<eGate>/Server/Registry/repository/ei_load/runtime/monk_scripts/ui**.

e*Ways

The validation phase includes seven e*Ways. One e*Way processes the input data file, five e*Ways write the records to the database and forward errors back to the IQ, and one e*Way writes the error records to an output file. The e*Ways use a subscriber pool to process the records, making the process faster and more efficient. The configuration files for the **ewBAD** and **ewLOAD_IN** e*Ways are located in **/<eGate>/Server/registry/repository/ei_load/runtime/configs/stcewfile**. The configuration files for the **ewLOAD##_OUT** e*Ways are located in **/.../ei_load/runtime/configs/stewcgenericmonk**. The e*Ways include:

- **ewBAD**
  This e*Way processes the records that cannot be loaded into the database. It subscribes to the **etBAD** event type, and writes the error records to an output file.
- **ewLOAD_IN**
  This e*Way processes records from the input data file (which is the "good" data file produced during the validation process). It writes the good data to the IQ. It subscribes and publishes to the **etLOAD** event type.
- **ewLOAD##_OUT**
  There are five e*Ways designed to process the records from **ewLOAD_IN** into the e*Index database. These e*Ways also write error records to the

IQ, which are then picked up by **ewBAD**. They subscribe to the **etLOAD** and **etBAD** event types, and publish to both event types as well. All five e*Ways use the same configuration file, so configuring one e*Way will configure them all.

Monk Library

Three monk files are included in the validation schema. These files are located in the schema directory in **/runtime/monk_library/ui**.

- **ui-load-person.monk**
  This file defines how records are processed into the e*Index database, including searching for possible matches, flagging potential duplicates, updating existing records, and inserting new records. You can modify this file to suit your requirements. SeeBeyond recommends that this only be done by someone very familiar with e*Index e*Ways and processing.
- **ui-stdver-eway-funcs.monk**
  This file contains the initialization and database connectivity commands that are used to connect to the e*Index database. The functions included in this file are described in detail in chapter 4 of your *e*Index Global Identifier Technical Reference*, and chapter 3 of that guide explains how each command is used in the e*Ways.
- **ui-custom.monk**
  This file contains the commands that create demographic, alias, transaction, address, and telephone information lists, which are passed as parameters to the Monk APIs. You may need to customize this file using the e*Index Monk functions provided in **ui-fns.monk** (for more information about the functions in **ui-fns.monk**, see Chapter 5 of the *e*Index Global Identifier Technical Reference*).

Load Schema Customizations

Most of the customizations made for each site are in the validation schema of the initial load, however you may need to modify the load schema as well. The components that you may need to modify for the load schema include the following.

Event Type Definitions

The only ETD that may require modification is **etdGOOD.ssc**. You should modify this file to match the **etdGOOD.scc** file from the validation schema (or you can just copy the validation schema file to the load schema). As mentioned before, SeeBeyond recommends that you only modify the **etdGOOD.ssc** ETD to match the ETD currently used by the production **.dsc** scripts. This allows the same script to be used as in the production environment, ensuring that both production and load data are processed in

the same manner. If you change **etdGOOD.ssc** you must also change the **ui-load.dsc** script.

Collaboration Rules

The schema provides a default collaboration rules file, **ui-load.dsc**. If you modify the ETDs for the schema, you need to modify this file as well. While this script performs additional checks required for the load process, the basic processing rules should be the same as those contained in the production script, **uidb.dsc**.

e*Ways

You can create additional e*Ways to load data into the database in order to maximize throughput. Since each of these e*Ways modify the same set of tables in the database, the throughput is limited by any concurrency limits imposed by the database and by what the database engine can support. You should not need to modify the configuration files for **ewLOAD_IN** and **ewBAD**, but you may want to modify the Start Up, Advanced, and Security options for the e*Ways. You need to enter information about the database in the Database Setup parameters for the **ewLOAD##_OUT** e*Ways.

Monk Scripts

The Monk files do not require modification, but you may want to customize the processing rules in **ui-load-person.monk**. This should only be done by someone who is very familiar with the Monk scripting language, e*Way processing, and your processing requirements. You may also need to modify the Monk lists defined in **ui-custom.monk** to make sure the fields in each list are formatted as required. See chapter 5 of the *e*Index Global Identifier Technical Reference* for more information.

Dropping Indexes

Before running the load schema, you may want to drop some of the indexes in the e*Index database in order to speed up processing time. While the indexes you can safely drop depend on the modifications made to the configurable query (in e*Index Administrator), with the default implementation you can safely drop all non-unique indexes from these tables: *ui_alias*, *ui_address*, *ui_local_id*, *ui_phone*, *ui_aux_id*, *ui_person_history*, *ui_alias_history*, *ui_address_history*, *ui_local_id_history*, *ui_phone_history*, and *ui_aux_id_history*. In addition, you can drop these indexes from the *ui_person* table: **idx_ui_person5**, **idx_ui_person6**, **idx_ui_person8**, and **idx_ui_person9**. The code for recreating these indexes is in the file **create_ui_tables.sql**, which is located in the path in which the database files were installed for e*Index in **\server\UIinitial**.

If you modified the configurable query, make sure you do not delete any indexes on the columns selected for either query. For more information about optimizing the initial load procedure and dropping indexes, see your SeeBeyond representative.

A Note About the Audit Trail

Performing an initial load after the e*Index database has been active and contains live data may cause some anomalies in the audit trail. No information should be lost, but changes to a member profile appear by date in the audit trail. This means that if an initial load was performed today that caused a profile to be updated, the updates may appear earlier in the audit trail because the create date or update date of the initial load record may be earlier than today. For example, member profile 100-000-0011 is added on 01/01/1980, and is updated through the GUI on 01/01/2000. On 01/01/2002 data is loaded through the initial load process, and the member profile is updated by a record whose last update date is 01/01/1990. In the resulting audit trail, the changes to profile 100-000-0011 appear in this order:

1. Member profile is added on 01/01/1980
2. Member profile is updated by initial load 01/01/1990 (even though the actual profile was updated in the e*Index database on 01/01/2002)
3. Member profile is updated through the GUI on 01/01/2000

Loading the Data File

Overview

Once you have validated the data to be loaded into the database, you must complete the following steps to load the data:

- Step 1: Copy the Data File
- Step 2: Configure the Database
- Step 3: Modify the Monk Files
- Step 4: Modify the ETDs
- Step 5: Modify the Collaboration Script
- Step 6: Configure the e*Ways
- Step 7: Stop the Archive Log
- Step 8: Load the Data
- Step 9: Fix any Records in the Error File
- Step 10: Restart the Archive Log
- Step 11: Run Database Reports

Note: *If you want to use additional e*Ways to process the data, you should create those e*Ways prior to beginning the following instructions.*

Step 1: Copy the Data Files

Once you complete the validation phase of the initial load process, you need to move the files containing the good data produced from the validation phase. The name and location of these files are specified in the configuration files of the **ewGOOD** e*Ways. If you did not change the defaults, the files are located in **/<eGate>/client/eIndex/load/output/*** and are named **good*.txt**.

▶ To copy the data files

Before you begin:

- ✓ Complete the validation process, as described in chapter 3 of this guide

1. Navigate to **/<eGate>/client/eIndex/load/output/a** directory in the e*Gate client environment.

2 Copy any files named **good*.txt** to **/<eGate>/client/eIndex/load/input**.

Note: *If you specified a different name for the "good" output file in the* ***ewGOOD_**** *e*Ways, you should copy the files with the name you specified.*

---

3 If you used more than one e*Way for the validation process, repeat steps 2 and 3 for any additional directories to which the e*Ways wrote "good" output files.

4 In the new location, change the **.txt** extension for all input files to **.dat**.

5 Continue to "Step 2: Configure the Database."

Step 2: Configure the Database

Before loading the member data into the database, you should make sure that systems and system codes with which the members are associated are already defined in the e*Index database. If any non-unique IDs are included in your initial load data, be sure to define the associated non-unique ID types as well. Finally, verify that the production version of the Vality rule set files is loaded into the database.

▶ To configure the database

Before you begin:

✓ Complete "Step 1: Copy the Data File"

1 Obtain system information from the source data, including the unique identification code for the system, a description or name for the system, and the format of the system's local identifiers.

2 Obtain non-unique ID type information from the source data, including the unique identification code for the ID type, a description or name for the ID type, and the format of the identifiers for each ID type.

3 Enter system information into e*Index using the System Maintenance function in the e*Index Administrator. System information is stored in the table *ui_facility*.

4 Enter non-unique ID type information into e*Index using the Non Unique ID Definition Maintenance function of e*Index Administrator.

5 Verify that the database contains the production version of the Vality rule set files. The easiest way to verify Vality information in the database is to launch the e*Index GUI, which downloads the database information into the GUI home directory. You can then compare the files in the GUI home directory against the production files to verify that the database contains the correct version.

6 Continue to "Step 3: Modify the Monk Files."

Step 3: Modify the Monk Files

You should not need to modify the Monk files provided with the load schema, but you can customize the files if needed. The files are located in **/<eGate>/server/registry/repository/ei_load/ runtime/monk_library/ui**.

The **ui-load-person.monk** file helps define how records are processed into the e*Index database. The collaboration rules script **ui-load.dsc** calls **ui-load-person.monk**. **ui-stdver-eway-funcs.monk** defines database connectivity and e*Way startup functions. You should not need to modify this file. For information about modifying **ui-custom.monk**, see "Step 4: Modify **ui-custom.monk**" in chapter 3 of this guide.

Notes:

- *These files should only be modified by someone who is very familiar with the Monk scripting language and with the data processing requirements of your system.*
- *Remember to commit the files to the sandbox and promote them to runtime when you have finished modifying them. You can also use the **stcregutil** command to do this.*

Step 4: Modify the ETDs

If you modified the ETD **etdGOOD.ssc** in the validation schema of the initial load program, you should modify the same file in the load schema to match. You do not need to modify the generic ETD.

Important! *Remember that if you modify this file, you'll need to modify **ui-load.dsc** to match the changes.*

▶ To modify the ETDs

Before you begin:

✓ Complete "Step 3: Modify the Monk Files"

1 In the e*Gate Enterprise Manager or any text editor, open the appropriate ETD file (located in **/<eGate>/server/registry/repository/ei_load/ runtime/monk_scripts/ui**).

2 Make the necessary modifications to the defined data structure.

Caution! *You should be very familiar with the Monk scripting language and the format of the incoming data file before modifying this file.*

---

3 Save and close the ETD file.

Note: *If you edited this file from the e*Gate Enterprise Manager, remember to commit the file to the sandbox, and then promote it to runtime. You can also commit the file using the* ***stcregutil*** *command.*

---

4 Continue to "Step5: Modify the Collaboration Script."

Step 5: Modify the Collaboration Script

The collaboration script file is where you specify how records are processed into the e*Index database. It is similar to the production collaborations rule file, **uidb.dsc**. If you modified the **etdGOOD.ssc** ETD, you need to change this file to match.

▶ To modify the collaboration script

Before you begin:

✓ Complete "Step 4: Modify the ETDs"

1 In the e*Gate Enterprise Manager or any text editor, open **ui-load.dsc** (located in **/<eGate>/server/registry/repository/ei_load/runtime/monk_scripts/ui**).

Caution! *You should be very familiar with the Monk scripting language and the format of the incoming data file before modifying this file.*

---

2 Make any necessary modifications to the collaboration rules.

3 Save and close the collaboration script file.

Note: *If you edited this file from the e*Gate Enterprise Manager, remember to commit the file to the sandbox, and then promote it to runtime. You can also commit the file using the* ***stcregutil*** *command.*

---

4 Continue to "Step 6: Configure the e*Ways."

Step 6: Configure the e*Ways

You may need to make some minor modifications to the e*Ways, depending on how you want to use the performance testing options, where you placed the input data file, and where you'd like the output files to be written. You need to specify database information for the **ewLOAD##_OUT** e*Ways.

Notes:

- *By default, all of the **ewLOAD01_OUT** e*Ways are configured to automatically start, the remaining "load out" e*Ways are not. If you want all of your e*Ways to start automatically when you open the e*Gate Monitor, you should configure them to do so.*
- *The "load out" e*Ways all use the same configuration file. You only need to modify the configuration file for one e*Way in order to change all the "load out" e*Ways.*

To configure the e*Ways

Before you begin:

✓ Complete "Step 5: Modify the Collaboration Script"

1 To configure the e*Way that reads the input files, do the following for the **ewLOAD_IN** e*Way.

- In the e*Gate Enterprise Manager open the Properties window for the e*Way.
- Modify the Startup, Advanced, and Security options as needed.
- Modify the Configuration file as needed.
  - You do not need to modify the General Settings or the Outbound (send) settings, but you can modify the Performance Testing parameters as needed.
  - If you did not copy the input data files into the default directories, or you did not give them a **.dat** extension, modify the Poller (inbound) settings by entering the new pathname in the **PollDirectory** parameter or entering the new file extension in the **InputFileMask** parameter. Modify the remaining parameters as needed.

2 To configure the e*Ways that write the data to the database, do the following for each **ewLOAD##_OUT** e*Way.

- In the e*Gate Enterprise Manager open the Properties window for the e*Way.
- Modify the Startup, Advanced, and Security options as needed.
- Modify the Configuration file as needed.
  - You do not need to modify the General Settings, Monk Configuration, or Communication Setup parameters, but you can customize these parameters if needed.
  - Modify the Database Set Up parameters by selecting the Database Type, and entering the Database Name, User Name, and Encrypted Password.

3 To configure the e*Way that writes the error records to the output data file, do the following for the **ewBAD** e*Way.

- In the e*Gate Enterprise Manager open the Properties window for the e*Way.
- Modify the Startup, Advanced, and Security options as needed.
- Modify the Configuration file as needed.
  - You do not need to modify the General Settings or the Poller (inbound) settings, but you can modify the Performance Testing parameters as needed.
  - If the path in which you want to store the output files is not the default path in **/<eGate/client**, modify the Outbound (send) settings by entering the new pathname in the **OutputDirectory** parameter. You can also modify the file name for the **OutputFileName** parameter. Modify the remaining parameters as needed.

4 Continue to "Step 7: Stop the Archive Log."

Step 7: Stop the Archive Log

In order to speed up the processing of the initial load, you may want to stop the Oracle archive log. You can monitor the load process through the e*Gate Control Broker. Also, at this time you may want to drop certain indexes or triggers from the database in order to speed up processing. For more information, see "Dropping Indexes" on page 4-7.

Step 8: Load the Data

Once you have made the necessary modifications to the schema files and the database environment, you are ready to run the load schema to insert the data into the e*Index database.

▶ To load the data

Before you begin:

- ✓ Complete "Step 7: Drop Non-unique Indexes"
- ✓ Make sure the input files are in the appropriate directories with the correct file names (as defined in the e*Way configuration file).

1 Start the control broker for the ei_load schema using the **stccb** command.

2 Start the ei_load schema in the e*Gate Monitor. If you have configured all of the e*Ways you are using to start automatically, they will start on their own. Otherwise, you need to start them manually.

3 As the data is loaded, you can monitor the process through the e*Gate Monitor, and by querying the ui_person table for the number of records processed (type **select count(*) from ui_person** at the SQL prompt).

4 As the input files are processed, any error record files are placed in **/<eGate>/client/eIndex/load/output** (if you used the default configuration).

5 When the process has finished running, check in the output file directory for any error records.

If error records are found, continue to "Step 9: Fix Records in the Error File."

If no error records are found, continue to "Step 10: Restart the Archive Log."

Step 9: Fix Records in the Error File

If any records were not processed, those records appear in the output file you specified in the configuration file for **ewBAD**. You can edit these records and then rerun the load schema against the fixed records.

▶ To fix records in the error file

Before you begin:

✓ Complete "Step 8: Load the Data"

1 Open the file you specified as the output file in the configuration file for **ewBAD**. You can edit this file in any text editor.

2 View the erred records, and make any necessary modifications to correct the errors.

3 Save the file to a new name to use as your new input file, using a **.dat** extension (or the extension you specified for the "load in" e*Ways).

4 Repeat "Step 8: Load the Data."

5 Repeat steps 1 through 4 in this procedure until all records have been processed.

6 Continue to "Step 10: Restart the Archive Log."

Step 10: Restart the Archive Log

When the load process is complete, and you have verified the data, remember to restart the Oracle archive log. Also, if you dropped any database indexes or triggers before running the initial load, recreate them now.

Step 11: Run Database Reports

To check on the state of the data processed into the database during the initial load, you can run reports to check for potential duplicates, assumed matches, and so on. These reports should have been installed with the e*Index database files, and should be located in **/<database_path>/server/UIreports/Production**. If you cannot locate these files, you can always reinstall them from the e*Index installation CD-ROM.

Use these reports to view potential duplicate records, to identify default values used in the data (such as 01/01/XXXX for an unknown birth date), and so on. You can then easily resolve the potential duplicates using the e*Index GUI on the Quality Workstation. The database reports may need to be customized for the initial load records. For more information about running reports, see your *Working with Reports for e*Index Global Identifier*.