*e\*Index Global Identifier Product Suite*

# e\*Index™ 4.1.1 to 4.5.2 Upgrade Guide

*Version 4.5.2*

**SEEBEYOND**

# Table of Contents

# Introduction

## About this Chapter

### Overview

This guide describes the process of upgrading an e*Index 4.1.1 environment to version 4.5.2. This introduction provides information about this guide, its organization, and the conventions used. It also includes a list of e*Index documents that may contain related information.

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.

| | |
|---|---|
| **Welcome** | Learn where to start in this guide if you are a new or experienced user |
| **About this Guide** | Learn how to use this guide, about its structure, and the conventions used |
| **Additional Resources** | Learn about other e*Index publications you may wish to review |

# What's Inside

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

# Introduction

## Welcome

Welcome to e*Index, SeeBeyond's enterprise-wide master person index. This document explains how to upgrade an e*Index 4.1.1 environment to version 4.5.2 on both the client and server machines, including the e*Index schema files, e*Index database and reports, and e*Index GUIs. This guide only describes how to upgrade from version 4.1.1. If you are currently at a later version, refer to the *e*Index Global Identifier Upgrade Guide* for version 4.5.2. The differences in database structure between the versions 4.1.1 and 4.5.2 are extensive, making this upgrade a complex procedure. Make sure to read this guide thoroughly before attempting an upgrade of an e*Index 4.1.1 database.

Chapter 2 of this document provides background information you should know before beginning the e*Index upgrade. In addition, the overview sections provided at the beginning of each chapter and at the beginning of each section within a chapter 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.

## To New Users

If you are new to e*Index 4.5.2 and have never performed an upgrade from version 4.1.1, you should browse through this guide and the *e*Index Global Identifier 4.1.1 to 4.5.2 Release Bulletin* before you begin the upgrade procedure. Please pay particular attention to the conceptual information provided in chapter 2 and at the beginning of each chapter. These sections are designed to provide background and explanatory information you may need to understand. After reading this overview information, you will be ready to perform an upgrade of your e*Index database.

## To Established Users

If you are a more advanced e*Index user, you may prefer to use this guide as a quick reference to find forgotten information about forgotten or to refresh your memory of the upgrade procedure. If you know what you need to do, but can't remember exactly how to do it, you can easily find what you need in the Table of Contents. Or, you can browse through the guide and find the appropriate background information or procedure by scanning headings and titles.

# About this Guide

## Purpose

This guide provides the information and instructions you need to upgrade an e*Index environment from version 4.1.1 to version 4.5.2. It includes background information and instructions for installing the upgrade files and performing the upgrade. This guide was designed to be used in conjunction with the *e*Index Global Identifier 4.1.1 to 4.5.2 Release Bulletin* so you can better understand the differences between the two versions and the customizations you may need to make to your new 4.5.2 environment.

## Scope

This guide provides step-by-step instructions for upgrading all of the components of e*Index from version 4.1.1 to 4.5.2, including the e*Index schema files, the database, reports, and the GUI. It includes navigational information, functional instructions, and background information where required. This guide also provides an illustration of the files that are installed into the e*Gate environment when you upgrade the e*Index components and the files that are installed for the database migration package (which is performed using Java).

This guide does not include information or instructions on using any of the e*Index applications. These topics are covered in the appropriate user guide (for more information, see "Additional Resources" on page 1-9).

## Intended Audience

Any user who will help perform the upgrade from version 4.1.1 to 4.5.2 should read this guide. The user should have a thorough understanding of the e*Index 4.1.1 database and Schema, and be familiar with the changes made for version 4.5.2. It is presumed that the reader of this guide is familiar with database administration, the database platform of the e*Index database, and the operating system(s) on which the e*Index database and schema run. Familiarity with the Java SDK environment is helpful, but not required.

## Using this Guide

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.

## Document Organization

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

| Chapter | Topics |
|---|---|
| Chapter 1, Introduction | ■ Introduction<br>■ About this Guide<br>■ Additional Resources |
| Chapter 2, Upgrade Process Overview | ■ About the Upgrade Procedure<br>■ Database Migration Methodology<br>■ Database Migration Considerations<br>■ e*Index 4.5.2 Database Model |
| Chapter 3, Upgrading the e*Index Schema Files | ■ Learning About the Upgrade Process<br>■ Performing the Upgrade |
| Chapter 4, Upgrading the GUIs and Publications | ■ Learning About the GUIs and Publications<br>■ Performing the Upgrade |
| Chapter 5, Upgrading the Database | ■ Learning About Migration Files<br>■ About Processing or Data Errors<br>■ Learning About the Migration Package Installation<br>■ Installing the Migration Package<br>■ Creating the Migration Databases<br>■ Migrating the Database |
| Chapter 6, Upgrading the Reports | ■ Learning About the e*Index Reports<br>■ Performing the Upgrade |

# Conventions

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

## Typographic Conventions

The following typographic conventions are used in this and other e*Index publications.

| Item | Convention | Example |
|---|---|---|
| Book titles | Title caps, italic | See the *e\*Index Global Identifier User's Guide* |
| Button names, key names, and key combinations | Bold | OK button<br><br>F1 key<br><br>Alt+Shift+V key combination |
| Chapter titles (and section titles within chapters) | Title caps, in quotation marks | See Chapter 2, " Upgrade Process Overview"<br><br>See "Database Migration Considerations" later in this chapter |
| New terms | Italic | A set of *Monk lists* is predefined to help you perform a variety of functions. |
| Menu names and commands | Bold, capitalization is identical to the interface | **Functions** menu<br><br>**Save** command |
| Typed command syntax | Bold for constants<br><br>Angle brackets denote user-supplied variables<br><br>Brackets denote optional values | Type **sqlplus <un>/<up>@<db_name> @disable_triggers** |
| Window, page, and dialog titles | Title cap | Select Components window<br><br>Choose Folder dialog |

## 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.

# 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 the system parameters for e*Index to meet your business requirements.  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 applications, 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.

- *e*Index Initial Load User's Guide*
  Provides the background information and instructions that system and database administrators need in order to load legacy data into the e*Index database, including a description of the expected data format and the schema files included with the load program.

- *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.

# Upgrade Process Overview

## About this Chapter

### Overview

This chapter presents the background information and provides a summary of the process for upgrading an e*Index 4.1.1 environment to version 4.5.2.

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.

| | |
|---|---|
| **Upgrade Overview** | Read an overview of the requirements for the upgrade and the process for upgrading the system |
| **Migration Methodology** | Learn background information about how the database migration is performed |
| **Migration Consideration** | Learn about issues to consider before migrating the database to version 4.5.2 |
| **Data Model** | View a data model of the e*Index 4.5.2 database structure |

## What's Inside

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

# About the Upgrade Procedure

## Overview

This section of the chapter outlines the requirements for e*Index 4.5.2 and for the upgrade package.  It also summarizes the order of the steps you need to follow to upgrade your e*Index environment from version 4.1.1.  If you are currently running a later version of e*Index, you should upgrade your environment using the information and instructions provided in the *e*Index Global Identifier Upgrade Guide*.

## Requirements for the e*Index 4.1.1 to 4.5.2 Upgrade

The installation of the upgrade files must be performed on a computer running Windows 95, Windows 98, Windows NT 4.0 with SP4 or later, or Windows 2000.  If you will be performing the actual upgrade on a Unix machine, you can either map a network drive to the Unix server from your installation machine and install the files directly into your Unix environment or you can install the files on the Windows machine and copy them to your Unix environment.  Instructions are provided for both methods.

Prior to beginning the installation, verify that you have the following requirements in place.

### e*Gate™ Integrator Requirements

If you are not currently running the following versions of e*Gate and the Database e*Way™, you need to upgrade them before performing the e*Index installation.

- e*Gate Integrator version 4.5.0 or later
- The Oracle e*Way version 4.5.0 or later
- HL7 Templates version 4.5.0 or later (only if you are transmitting HL7 messages)
- e-Mail e*Way 4.5.0 or later (only if you will be using the Event Notification functionality of e*Index)

### Database Platform Requirements

The e*Index database is supported on Oracle 8.1.7.  An Oracle database requires Oracle Server on the database server machine, and Oracle Client on the client workstations and the e*Gate server machines on which you install the e*Index schemas.  For an Oracle database server, you should have at an absolute minimum 256 MB of RAM.  SeeBeyond recommends installing or

upgrading to Oracle version 8.1.7.2.1, which provides a smoother export and import of the e*Index database.

## Operating Systems

The e*Index database can be installed on any operating system supported by Oracle.  The e*Index schema files can be installed on any of the following platforms:

- Windows NT 4.0, SP4 or later
- Solaris 2.6 or later
- HPUX 11 or 11i
- AIX 4.3.3 or 5.1
- TRU64 4.0f and 5.0a

The e*Index GUIs can be installed on any of the following platforms:

- Windows NT 4.0, SP4 or higher
- Windows 95
- Windows 98

## Java Requirements

The machine from which you run the database migration must have Java™ 2 SDK, Standard Edition 1.1.8 or later installed.  For HP Unix 10.20, you must install version 1.1.8; for HP Unix 11.0, you must install version 1.2 or later; for all other platforms, version 1.3.1 is recommended.

# Installation Overview

Most of the setup for the e*Index upgrade is performed using standard InstallShield® Wizards with specific customizations for each component. The database components are provided in a separate InstallShield from the e*Gate, GUI, and report components.  Each step outlined below describes how to install one component, and is described in detail in its own chapter in this guide.  Once you have installed the files for a specific step, some modifications may be required.  All required modifications are described in the appropriate chapter.

- Upgrade the e*Index Schema files (described in Chapter 3)
- Upgrade the GUI (described in Chapter 4)
- Upgrade the database (described in Chapter 5)
- Upgrade the reports (described in Chapter 6)

## Upgrade the e*Index Schema Files

Chapter 3, "Upgrading the e*Index Schema Files", outlines the steps required to upgrade the e*Index Schema components in the e*Gate environment.  This chapter discusses file structure and the modifications that may be required to the Schema files in order to process data in the same manner as in version 4.1.1.  If you have customized your e*Index 4.1.1 Schema, similar modifications will be required for version 4.5.2.

## Upgrade the GUIs

Chapter 4, "Upgrading the GUIs", contains the instructions for upgrading the e*Index GUIs on the client workstations.  This chapter includes hardware and software requirements for the installation.  If you have not already copied the e*Index publications to your computer, you can also copy the new e*Index Electronic Library.  This library includes the suite of e*Index documentation and a Welcome document to help you navigate through the files.

## Upgrade the Database

Chapter 4, "Migrating the Database", provides the instructions you need to upgrade the e*Index database from version 4.1.1 to 4.5.2.  This is a complex task due to the database changes between 4.1.1 and 4.5.2, and involves migrating your production data from a copy of your 4.1.1 database to a new 4.5.2 database.  The database structure for the 4.5.2 database is provided so you can verify the new database.

## Upgrade the Reports

Chapter 6, "Upgrading the Reports", contains the instructions for upgrading the standard production reports for version 4.5.2.  Because of the database changes between version 4.1.1 and 4.5.2, you will need to modify the new reports with any customizations you made to the earlier versions.  In addition, any custom reports you created for e*Index will need to be updated.

# Database Migration Methodology

## Overview

This section of the chapter provides information about how data is migrated during the database upgrade, and what issues you should consider before beginning the upgrade process.

## Migration Methodology

The migration process involves three databases: the production database, an e*Index 4.1.1 source database, and an e*Index 4.5.2 target database.  To begin the upgrade process, the  data from the production database is exported and then imported into the 4.1.1 source database to create an exact replica of the production database.  After this, the production database is not touched for the remainder of the migration and can continue to be used as the production database.  The data is migrated from the 4.1.1 source database to the 4.5.2 target database.  The 4.5.2 target database becomes the production database once the upgrade process is complete, and you have verified that no errors occurred.  The diagram below illustrates the flow of data during the upgrade process.

```
PRODUCTION SERVER

   Production 4.1.1
   Database

        │
        │ Export
        │ data to
        │ file
        ▼

Production Data                    UPGRADE PROCESS SERVER
Export File
        │
        │                Source 4.1.1     Migrate data        Target 4.5.1
        └──────────────▶ Database     ──────────────▶         Database
          Import data to             from source to
          source database            target database
```
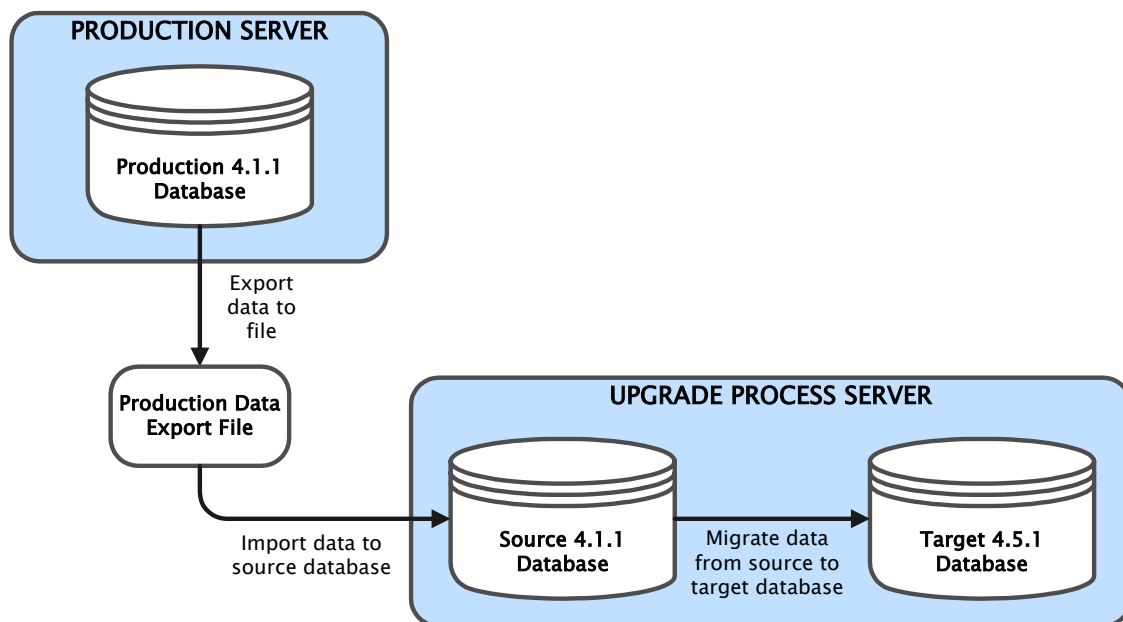
## About e*Gate Queuing

When you export your production database, you need to queue any incoming transactions to the 4.1.1 Schema to avoid losing any information.

At this time, you will also being queuing the incoming transactions to the 4.5.2 Schema.  During the migration process, you can continue to use the e*Index 4.1.1 production database to process daily transactions, but at the same time you need to continue queuing those transactions in the 4.5.2 Schema.  Then, when the 4.5.2 database is ready to move into production, you can start processing the stored Events from the 4.5.2 queue to bring the database current.

## Migration Phases

Installing the migration package files creates SQL scripts and Java files that you will use to perform the upgrade process.  The migration is performed in phases.

**1**  Migrate the security and common table data

**2**  Migrate the auxiliary table data

**3**  Migrate the person data

**4**  Migrate the assumed match table

You can perform each of these phases in individual steps.  For example, when you migrate the security and common table data, you can migrate the security data in one step; races, religions, suffixes, and languages in a second step; and the remaining common tables in the final step.  When you migrate person data, you can specify a start and end date for the records to be migrated, allowing you to perform the migration in batches.  It is important to remember that the person records must be migrated in chronological order.  These phases are all performed using Java commands.

## How Person Data is Migrated

When you migrate person information from the 4.1.1 source database to the 4.5.2 target database, all of the data comes from the *ui_history* table since that table maintains all history and current records for each person.  The history of changes to person data will be migrated into a complete set of 4.5.2 history records, with the most current information being migrated into *ui_person*.  In addition, since the *ui_person* table no longer stores address and telephone information, address information will be migrated into *ui_address* and *ui_address_history*, and telephone information will be migrated into *ui_phone* and *ui_phone_history*.  You can specify the address and phone type to assign to each address and telephone number in the migration properties file (this file is described in chapter 5).  The migration process uses the timestamp of the records in the 4.1.1 history table to determine the transaction number to assign to the records migrated into the 4.5.2 database.  This method maintains the sequence of transactions for each person record.

## Error Handling

Information about any record that cannot be migrated is stored in exception tables in the source database.  These tables identify the UID of the records that were in error and provide a brief description of the errors.  You can then repair the data and reprocess the exception records.  Once an error is found for a specific UID, no transactions associated with that UID can be processed and any subsequent transactions associated with an UID record with errors will also appear in the exception tables until the error is fixed.  If you run the migration in phases, you should check the exception tables and reprocess data after each phase.

## Data Mapping for the Migration

Because of the differences in database structure between versions 4.1.1 and 4.5.2, data from the 4.1.1 database will be migrated into some different tables in the 4.5.2 database.  The most notable difference is that person data is now stored in several different tables.  Person data will be migrated as follows:

■ Data stored in *ui_person* will be migrated to *ui_person*, *ui_address*, *ui_phone*, *ui_person_x_name*, and *ui_mrg_trans*.

■ Data stored in *ui_history* will be migrated to *ui_person_history*, *ui_address_history*, *ui_phone_history*, *ui_phone*, *ui_address*, and *ui_mrg_trans*.

■ Data stored in *ui_local_id* will be migrated to *ui_local_id* and *ui_local_id_history*.

■ Data stored in *ui_alias* will be migrated to *ui_alias*, *ui_alias_history*,  and *ui_alias_x_name*.

Information from each table above is also stored in the new *ui_transaction* table.  For information about how administrative tables are migrated, see appendix A of this guide.

# Database Migration Considerations

## Overview

This section of the chapter provides background information about the data migration process. Because the database structure differences between the two versions are so great, data cannot be transferred directly into the new database. This section outlines the issues you should consider before beginning the data migration to better prepare for the process.

## e*Index Security Considerations

Between e*Index versions 4.1.1 and 4.5.2, the e*Index Security application has been redesigned. This redesign makes security for e*Index easier to configure and more robust than before. Due to the changes, once you migrate the security tables you will need to make modifications to several profiles and possibly recreate some profiles. You can choose not to migrate the security tables and instead redesign your security configuration and recreate the security setup completely.

### User Profiles

User profiles are migrated as completely as possible with the following conditions:

- The UI user is created as an Administrator user, and is the only user created as an Administrator user. This user is assigned to the three default user groups, which grants all access permissions. If any user profiles need to be recreated, the UI user needs to perform that function.

- All e*Index GUI access permissions assigned to a user profile or user group are migrated as completely as possible. See "Terminology Changes" under "e*Index Security GUI Enhancements" in the *e*Index 4.1.1 to 4.5.2 Release Bulletin* for a list of changes in access permissions between versions 4.1.1 and 4.5.2. Note that the migration package migrates e*Index GUI access permissions whose names changed, but you need to grant the access permissions that are new to e*Index 4.5.2.

- Super Users are migrated as Regular users.

- Since there are no Group Supervisor users in version 4.5.2, any existing Group Supervisors are migrated as Regular users, meaning they can no longer create user profiles.

- Regular users are all migrated as Regular users.

- All user profiles retain assignments to user groups.

## User Groups

User groups are migrated in their entirety, and their access permissions are migrated as completely as possible.  Each user group retains all user profiles that are assigned to it.

## Passwords

After migration, the **User Must Change Password at Next Logon** checkbox will be selected for all users except the UI user.  Passwords cannot be migrated, so users must log on using their existing user name and their user name **in upper case** as the password.  The Change Password dialog will appear the first time each user logs on to the upgraded system.

*Note: Make sure users log on using an upper case password.  Otherwise they will be unable to modify their password and will be unable to log on.*

## Event Notification

The redesigned security application provides a new function that allows you to specify users to be notified when specific Events occur.  This function works in conjunction with the e-Mail e*Way to send e-Mail notifications to the specified user.  Before migrating the database, you should determine whether this function will be used in your e*Index 45.2 environment.

# e*Index Administrator Considerations

You can migrate any of the code table data created through the e*Index Administrator (previously e*Index Data Dictionary).  When you perform the migration, default data will exist in the 4.5.2 target database and in many cases, the default data has been improved (for example, spelling has been corrected in the Country and Religion tables).  If you have made no modifications to a code table, you do not need to migrate that table since the existing 4.5.2 tables will be populated with improved default data.  For all code tables you migrate, the processing codes and their corresponding descriptions are migrated in their entirety.

## Terminology Changes

One issue to consider when deciding which administrator tables to migrate is the terminology changes between versions 4.1.1 and 4.5.2.  The table below outlines the changes.

| e*Index 4.1.1 | e*Index 4.5.2 |
|---------------|---------------|
| Facility | System |
| System | Source |
| Patient | Person |

The two tables affected most by this change are *ui_comments* and *ui_message*. When you migrate the comments table, the terminology is not updated for existing comments, so it is possible that at some point records in the new 4.5.2 database will contain comments that use both sets of terminology. SeeBeyond recommends migrating the comments table; otherwise all comments associated with person records will be lost.

The *ui_message* table contains the application messages that appear when you use the e*Index GUI.  Unless you have made a large number of modifications to the data in this table, SeeBeyond recommends that you do not migrate this table since the old terminology may cause confusion for GUI users.

## Control Keys

If you migrate the *ui_control* table, both the description and the value associated with a specific control key are overwritten.  There are several new control keys that you need to configure manually.  Before you begin the migration, analyze the requirements for these new parameters so you can customize them during the migration.  One important change in control key functionality is that the EXTNSVSRCH control key now only controls GUI alphanumeric searches, and not backend or phonetic queries (see "Extensive Searching" below for more information).  For more information about new and obsolete control keys, see "Control Key Changes" in the *e*Index 4.1.1 to 4.5.2 Release Bulletin*.

## Extensive Searching

If you have enabled extensive searching in your 4.1.1 database (that is, the EXTNSVSRCH control key is set to **Y**), then you are searching the alias table when processing records from both the GUI and the e*Ways.  In e*Index 4.5.2, the EXTNSVSRCH control key only affects alphanumeric searches from the GUI.  Phonetic searches and candidate selection searches are controlled through the configurable query (for more information about the configurable query, see chapter 5 of the *e*Index Administrator User's Guide*).  In order to perform extensive searching for all searches, you need to modify the configurable queries to query the alias table.  Instructions for modifying the configurable queries to enable extensive searching are provided in "Configuring Extensive Searching" in chapter 5 of the *e*Index Administrator User's Guide*.

## New Configuration Functions

e*Index 4.5.2 is much more customizable than e*Index 4.1.1, which means you need to analyze additional requirements before completing the migration.  For more information about the configuration functions listed below, see chapter 5 of the *e*Index Administrator User's Guide*.

■ **Country-specific Options**
Country-specific options allow you to define the attributes of certain e*Index components that are specific to the country specified by the COUNTRY control key.  These options include display options, such as the labels for the tabbed headings on the GUI windows; address-parsing options for the address rule set that corresponds to the country specified; and formatting options, such as the format and length of the national identifier and telephone numbers.

■ **Display Configuration**
e*Index 4.5.2 allows you to specify which fields will appear on the e*Index GUIs and the name of each field that appears on the GUIs.

■ **Configurable Query**
The configurable query defines how phonetic searches are processed and how e*Index searches for possible matches through the GUI and the e*Ways.  By default, this query is configured to process searches in the same manner as in e*Index 4.1.1.  If you want to continue processing records in the same manner, you only need to change the configurable query if extensive searching is enabled (see "Extensive Searching" earlier for more information).

# Database Customizations

If you have customized your e*Index database by creating any of the following items, it is important to note that they are not migrated.  If you want to maintain the functionality of these items, you need to add them to the new e*Index 4.5.2 database manually.

■ New database tables

■ New columns added to existing tables

■ New stored procedures

■ New triggers

■ New views

# Audit Trails

Events that trigger an audit trail record differ between e*Index 4.1.1 and 4.5.2, which may lead to some anomalies in the audit trail for certain records.  A complete audit trail for all records is maintained in both 4.1.1 and 4.5.2, however there will be differences between a record's audit trail in version 4.1.1 and the same record's audit trail in 4.5.2.  The 4.5.2 audit trail may have fewer images for certain records, but it more accurately reflects the actual history of the record, and especially of the associated local ID and alias changes.  Many of the differences are caused by changes to local ID and alias information to person records since the migration package uses update dates

and create dates to link information between *ui_history*, *ui_local_id*, and *ui_alias*.

## Potential Duplicates

It is important to note that the *ui_duplic* table is not migrated to the 4.5.2 database.  All potential duplicate records should be either resolved or merged prior to performing the migration in order to maintain the integrity of your production data.  If any rows remain in the *ui_duplic* table at the time of migration, an error will be logged.

## Merging and Unmerging Records

When production data is migrated from the 4.1.1 database to the 4.5.2 database, the merge and unmerge information for each record is retained.  One the migration is complete, any records that were merged in the 4.1.1 database can be unmerged in the 4.5.2 database if necessary.

## UID Numbers

The data migration retains all UID numbers assigned to each person record.  When the migration is complete, the next UID number (defined in the *ui_seq_no* table) is automatically set so UIDs continue to be assigned sequentially.

## Case-insensitive Searching

e*Index 4.1.1 introduced case-insensitive searches for the GUI.  In e*Index 4.5.2, searches are always case-insensitive.  You can also specify that the GUI allow mixed-case entry when entering or searching for person information in the GUI by changing the value of the MIXEDCASE control key to **Y**.

## Region-specific Security

Region-specific security is a new feature of e*Index, and allows you to limit a user's ability to view member information by the region in which their information originated.  Before you create your 4.5.2 target database, you need to determine whether you will use region-specific security.  This capability needs to be installed in the database separately (for more information, see chapter 3 of the *e*Index Global Identifier Upgrade Guide*).  Once you install region-specific security, regions need to be defined and assigned to systems, and user profiles must be assigned to regions.  For more information about this feature, see "Region-specific Security" in the *e*Index 4.1.1 to 4.5.2 Release Bulletin*.

## HL7 Messages

The ETD for e*Index 4.5.2 was updated to provide a more customizable format and to include a larger number of fields.  This means that certain fields, such as the create_function and update_function fields in the *ui_person* and *ui_history* tables do not exist in the 4.5.2 database.  To ensure that this information is not lost during the migration, the create_function or update_function for the 4.1.1 record will be inserted into the terminal_id field in the *ui_transaction* table.

## Vality Rule Set Files

e*Index 4.5.2 uses a set of rule set files in addition to the **UI** rule set to perform address parsing.  While the migration process uses address rule set files, it does not use the standard **UI** rule set since no matching is performed on names during the migration.  Instead, the migration package uses a rule set named **ENCODE**.  This rule set is used to create phonetic codes for person names.

Before you perform the data migration, analyze your matching algorithm requirements for address-parsing so you can customize the address rule set for both the production and migration environments.  You should use the same address rule set for both environments to maintain data consistency.  In addition, use your existing 4.1.1 **UI** rule set to customize the new 4.5.2 **UI** rule set. This rule set has been updated to Vality version 3.11, additional matching logic is provided, and the pattern-action file is more comprehensive.  For information about specific changes to the **UI** rule set, see "Vality Rule Set Modifications" in the *e*Index 4.1.1 to 4.5.2 Release Bulletin*.

Once you customize the e*Index 4.5.2 **UI** rule set, you need to load the new files into the database using the Rule Set Maintenance function of the e*Index Administrator.  If you choose to migrate the rule set tables (see "Step 7: Migrate Auxiliary Tables" in chapter 5), load the customized 4.5.2 files after the migration.  If you choose not to migration the rule set tables, load the customized files at any time before moving to production (however, SeeBeyond recommends loading the files following the migration to ensure that no production files are overwritten).

*Note:  SeeBeyond recommends that the same matching logic used in the 4.1.1 environment also be used in the 4.5.2 environment to maintain weighting consistency.*

# e*Index 4.5.2 Oracle Database Model

The diagrams on the following pages illustrate the table structure for e*Index version 4.5.2 for Oracle. Your actual database may vary slightly from this model depending upon the release you have installed. The *ui_person* and *ui_transaction* tables are displayed on two different pages to better illustrate the connections to these two tables.

where_clause_id = where_clause_id

**ui_cand_from_table**

| from_table_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| where_clause_id | <fk> | NUMBER(10) | not null |
| sql_table_id | <fk> | NUMBER(10) | not null |
| table_alias | | VARCHAR2(10) | null |
| sequence | | NUMBER(10) | not null |

ui_cand_from_table_1

**ui_cand_where_clause**

| where_clause_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| cand_sql_id | <fk> | NUMBER(10) | not null |
| where_clause_name | | VARCHAR2(40) | not null |
| where_clause | | LONG | null |

where_clause_id = where_clause_id

sql_table_id = sql_table_id

**ui_object_type**

| object_type | <pk> | VARCHAR2(5) | not null |
|---|---|---|---|
| description | | VARCHAR2(40) | null |

**ui_cand_sql_table**

| sql_table_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| table_name | | VARCHAR2(30) | not null |
| selected | | CHAR(1) | not null |
| readonly | | CHAR(1) | not null |

cand_sql_id = cand_sql_id

object_type = object_type

**ui_cand_where_column**

| where_column_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| where_clause_id | <fk> | NUMBER(10) | not null |
| sql_column_id | <fk> | NUMBER(10) | not null |
| sequence | | NUMBER(10) | not null |

**ui_cand_sql**

| cand_sql_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| object_type | <fk> | VARCHAR2(5) | not null |
| use_type | | CHAR(1) | not null |
| sql_name | | VARCHAR2(40) | not null |

sql_column_id = sql_column_id

cand_sql_id = cand_sql_id

**ui_cand_sql_column**

| sql_column_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| sql_table_id | <fk> | NUMBER(10) | not null |
| column_name | | VARCHAR2(30) | not null |
| column_type | | VARCHAR2(5) | not null |
| column_length | | NUMBER(10) | not null |
| attr_name | | VARCHAR2(40) | null |
| phonetic_source | | VARCHAR2(30) | null |
| phonetic_type | | CHAR(1) | null |
| match_column | | CHAR(1) | null |
| selected | | CHAR(1) | not null |
| readonly | | CHAR(1) | not null |

**ui_cand_select_column**

| select_column_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| cand_sql_id | <fk> | NUMBER(10) | not null |
| sql_column_id | <fk> | NUMBER(10) | not null |
| sequence | | NUMBER(10) | not null |
| match_column | | CHAR(1) | not null |

sql_column_id = sql_column_id

**dequeue_lock**

| sess_id | NUMBER | null |
|---|---|---|
| sess_activity | DATE | null |
| last_seq | NUMBER(20) | null |
| del_seq | NUMBER(20) | null |

**ui_monthly_stat**

| be_new_u_id | NUMBER | null |
|---|---|---|
| fe_new_u_id | NUMBER | null |
| be_merge | NUMBER | null |
| fe_merge | NUMBER | null |
| be_unmerge | NUMBER | null |
| fe_unmerge | NUMBER | null |
| be_deact | NUMBER | null |
| fe_deact | NUMBER | null |
| pot_dup | NUMBER | null |
| pot_dup_res | NUMBER | null |
| month | VARCHAR2(20) | null |
| year | NUMBER | null |

**ui_audit**

| ui_audit_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| user_id | | VARCHAR2(20) | not null |
| timestamp | | DATE | not null |
| uid1 | | VARCHAR2(15) | null |
| uid2 | | VARCHAR2(15) | null |
| funct | | VARCHAR2(40) | null |
| detail | | VARCHAR2(150) | null |
| pat_acct_num | | VARCHAR2(20) | null |
| terminal_id | | VARCHAR2(25) | null |
| facility | | VARCHAR2(20) | null |
| location | | VARCHAR2(15) | null |
| order_num | | VARCHAR2(20) | null |
| app | | VARCHAR2(10) | null |

**ui_seq_no**

| table_name | <pk> | VARCHAR2(30) | not null |
|---|---|---|---|
| seq_no | | NUMBER(15) | not null |

**ui_config**

| interface | <pk> | VARCHAR2(255) | not null |
|---|---|---|---|
| code | <pk> | VARCHAR2(255) | not null |
| value | | NUMBER | not null |

pk_idx_ui_config

**ui_control**

| ctrl_key | <pk> | VARCHAR2(10) | not null |
|---|---|---|---|
| description | | VARCHAR2(50) | null |
| ctrl_value | | VARCHAR2(30) | null |
| create_date | | DATE | null |

pk_idx_control

**ui_dept**

| dept_code | <pk> | VARCHAR2(5) | not null |
|---|---|---|---|
| description | | VARCHAR2(20) | null |
| date_time | | DATE | null |

pk_idx_dept

**ui_canned_msg**

| code | <pk> | VARCHAR2(5) | not null |
|---|---|---|---|
| description | | VARCHAR2(80) | not null |
| create_date | | DATE | null |

pk_idx_canned

**ui_message**

| code | <pk> | VARCHAR2(5) | not null |
|---|---|---|---|
| description | | LONG | not null |
| message_box_header | | VARCHAR2(50) | not null |
| icon | | VARCHAR2(15) | null |
| button | | VARCHAR2(20) | null |
| default_button | | NUMBER(1) | null |
| message_type | | VARCHAR2(8) | null |
| application | | VARCHAR2(10) | null |
| date_time | | DATE | null |

pk_idx_ui_message

**ui_zip**

| zip_code | <pk> | VARCHAR2(8) | not null |
|---|---|---|---|
| zip4 | | VARCHAR2(4) | null |
| city | <pk> | VARCHAR2(30) | not null |
| state | <pk> | VARCHAR2(10) | not null |
| county | | VARCHAR2(3) | null |
| residence_code | | VARCHAR2(4) | null |
| create_date | | DATE | null |

pk_idx_zip

**ui_comment**

| ui_comment_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| u_id | | VARCHAR2(15) | not null |
| type | | VARCHAR2(8) | not null |
| timestamp | | DATE | not null |
| comment_text | | LONG | null |
| ui_org | | VARCHAR2(15) | null |

ui_id_comment

**ui_msg_header**

| ui_msg_header_id | <pk> | NUMBER(20) | not null |
|---|---|---|---|
| queue_id | | CHAR(1) | not null |
| errors | | NUMBER(10) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

ui_msg_header_1

ui_msg_header_id = ui_msg_header_id

**ui_msg_detail**

| ui_msg_detail_id | <pk> | NUMBER(20) | not null |
|---|---|---|---|
| ui_msg_header_id | <fk> | NUMBER(20) | null |
| msg | | VARCHAR2(512) | not null |

fk_ui_msg_detail

**ui_person_history**

| ui_person_history_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| u_id | | VARCHAR2(15) | not null |
| person_cat_code | | VARCHAR2(8) | null |
| last_name | | VARCHAR2(40) | null |
| first_name | | VARCHAR2(40) | null |
| middle_name | | VARCHAR2(30) | null |
| suffix | | VARCHAR2(10) | null |
| title | | VARCHAR2(8) | null |
| dob | | DATE | null |
| death | | VARCHAR2(1) | null |
| sex | | VARCHAR2(8) | null |
| mstatus | | VARCHAR2(8) | null |
| ssn | | VARCHAR2(16) | null |
| race | | VARCHAR2(8) | null |
| ethnic | | VARCHAR2(8) | null |
| religion | | VARCHAR2(8) | null |
| language | | VARCHAR2(8) | null |
| spouse_name | | VARCHAR2(100) | null |
| mother_name | | VARCHAR2(100) | null |
| mother_mn | | VARCHAR2(40) | null |
| father_name | | VARCHAR2(100) | null |
| maiden | | VARCHAR2(40) | null |
| pob_city | | VARCHAR2(30) | null |
| pob_state | | VARCHAR2(10) | null |
| pob_country | | VARCHAR2(20) | null |
| vip_flag | | VARCHAR2(8) | null |
| vet_status | | VARCHAR2(8) | null |
| status | | VARCHAR2(8) | null |
| fname_phonetic_code | | VARCHAR2(8) | null |
| lname_phonetic_code | | VARCHAR2(8) | null |
| mname_phonetic_code | | VARCHAR2(8) | null |
| mother_mn_phonetic_code | | VARCHAR2(8) | null |
| maiden_phonetic_code | | VARCHAR2(8) | null |
| spouse_phonetic_code | | VARCHAR2(8) | null |
| mother_phonetic_code | | VARCHAR2(8) | null |
| father_phonetic_code | | VARCHAR2(8) | null |
| drivers_license | | VARCHAR2(20) | null |
| drivers_license_st | | VARCHAR2(10) | null |
| dod | | DATE | null |
| death_certificate | | VARCHAR2(10) | null |
| nationality | | VARCHAR2(8) | null |
| citizenship | | VARCHAR2(8) | null |
| pension_no | | VARCHAR2(15) | null |
| pension_exp_date | | DATE | null |
| repatriation_no | | VARCHAR2(16) | null |
| district_of_residence | | VARCHAR2(8) | null |
| lga_code | | VARCHAR2(4) | null |
| military_branch | | VARCHAR2(4) | null |
| military_rank | | VARCHAR2(4) | null |
| military_status | | VARCHAR2(4) | null |
| dummy_date | | DATE | null |
| class1 | | VARCHAR2(20) | null |
| class2 | | VARCHAR2(20) | null |
| class3 | | VARCHAR2(20) | null |
| class4 | | VARCHAR2(20) | null |
| class5 | | VARCHAR2(20) | null |
| string1 | | VARCHAR2(40) | null |
| string2 | | VARCHAR2(40) | null |
| string3 | | VARCHAR2(40) | null |
| string4 | | VARCHAR2(40) | null |
| string5 | | VARCHAR2(40) | null |
| string6 | | VARCHAR2(40) | null |
| string7 | | VARCHAR2(100) | null |
| string8 | | VARCHAR2(100) | null |
| string9 | | VARCHAR2(100) | null |
| string10 | | VARCHAR2(255) | null |
| date1 | | DATE | null |
| date2 | | DATE | null |
| date3 | | DATE | null |
| date4 | | DATE | null |
| date5 | | DATE | null |
| transaction_no | <fk> | NUMBER(10) | not null |

transaction_no = transaction_no

TO UI_TRANSACTION

FROM UI_PERSON_HISTORY

transaction_no = transaction_no

transaction_no = trans_no
transaction_no = unmrg_trans_no_from
transaction_no = trans_no_from
transaction_no = unmrg_trans_no
transaction_no = transaction_no
transaction_no = transaction_no
transaction_no = transaction_no
transaction_no = transaction_no

transaction_no = prev_transaction_no
transaction_no = transaction_no

**ui_transaction**

| | | | |
|---|---|---|---|
| transaction_no | <pk> | NUMBER(10) | not null |
| u_id | | VARCHAR2(15) | not null |
| system | | VARCHAR2(8) | null |
| dept | | VARCHAR2(8) | null |
| terminal_id | | VARCHAR2(25) | null |
| session_id | | VARCHAR2(15) | null |
| update_facility_id | | VARCHAR2(20) | not null |
| update_function | | VARCHAR2(8) | not null |
| update_date | | DATE | not null |
| update_userid | | VARCHAR2(20) | not null |

u_id = u_id

transaction_no = transaction_no

**ui_assumed_match**

| | | | |
|---|---|---|---|
| assumed_match_id | <pk> | NUMBER(10) | not null |
| u_id | <fk> | VARCHAR2(15) | not null |
| weight | | FLOAT | not null |
| prev_transaction_no | <fk> | NUMBER(10) | not null |
| transaction_no | <fk> | NUMBER(10) | not null |

**ui_aux_id**

| | | | |
|---|---|---|---|
| aux_id_id | <pk> | NUMBER(10) | not null |
| u_id | <fk> | VARCHAR2(15) | not null |
| aux_id_def | <fk> | NUMBER(10) | not null |
| id | | VARCHAR2(40) | not null |
| transaction_no | <fk> | NUMBER(10) | null |

u_id = u_id
aux_id_def = aux_id_def

**ui_aux_id_def**

| | | | |
|---|---|---|---|
| aux_id_def | <pk> | NUMBER(10) | not null |
| id_type | | VARCHAR2(8) | not null |
| id_name | | VARCHAR2(30) | not null |
| mask | | CHAR(1) | not null |
| id_length | | NUMBER(10) | not null |
| format | | VARCHAR2(40) | not null |
| variable_length | | char(1) | not null |

aux_id_def = aux_id_def

**ui_aux_id_history**

| | | | |
|---|---|---|---|
| aux_id_id | | NUMBER(10) | not null |
| u_id | <fk> | VARCHAR2(15) | not null |
| aux_id_def | <fk> | NUMBER(10) | null |
| id | | VARCHAR2(40) | null |
| transaction_no | <fk> | NUMBER(10) | not null |

u_id = u_id

**ui_address**

| | | | |
|---|---|---|---|
| address_id | <pk> | NUMBER(10) | not null |
| u_id | <fk> | VARCHAR2(15) | not null |
| address_type | | VARCHAR2(8) | not null |
| address1 | | VARCHAR2(40) | null |
| address2 | | VARCHAR2(40) | null |
| address3 | | VARCHAR2(40) | null |
| address4 | | VARCHAR2(40) | null |
| city | | VARCHAR2(30) | null |
| state_code | | VARCHAR2(10) | null |
| postal_code | | VARCHAR2(8) | null |
| postal_code_ext | | VARCHAR2(4) | null |
| county | | VARCHAR2(20) | null |
| country_code | | VARCHAR2(20) | null |
| house_number | | VARCHAR2(10) | null |
| street_dir | | VARCHAR2(5) | null |
| street_name | | VARCHAR2(40) | null |
| street_name_phonetic | | VARCHAR2(8) | null |
| street_type | | VARCHAR2(5) | null |
| transaction_no | <fk> | NUMBER(10) | null |

u_id = u_id

**ui_address_history**

| | | | |
|---|---|---|---|
| address_id | | NUMBER(10) | not null |
| u_id | <fk> | VARCHAR2(15) | not null |
| address_type | | VARCHAR2(8) | null |
| address1 | | VARCHAR2(40) | null |
| address2 | | VARCHAR2(40) | null |
| address3 | | VARCHAR2(40) | null |
| address4 | | VARCHAR2(40) | null |
| city | | VARCHAR2(30) | null |
| state_code | | VARCHAR2(10) | null |
| postal_code | | VARCHAR2(8) | null |
| postal_code_ext | | VARCHAR2(4) | null |
| county | | VARCHAR2(20) | null |
| country_code | | VARCHAR2(20) | null |
| house_number | | VARCHAR2(10) | null |
| street_dir | | VARCHAR2(5) | null |
| street_name | | VARCHAR2(40) | null |
| street_name_phonetic | | VARCHAR2(8) | null |
| street_type | | VARCHAR2(5) | null |
| transaction_no | <fk> | NUMBER(10) | not null |

u_id = u_id

**ui_person**

| | | | |
|---|---|---|---|
| u_id | <pk> | VARCHAR2(15) | not null |
| person_cat_code | | VARCHAR2(8) | null |
| last_name | | VARCHAR2(40) | not null |
| first_name | | VARCHAR2(40) | not null |
| middle_name | | VARCHAR2(30) | null |
| suffix | | VARCHAR2(10) | null |
| title | | VARCHAR2(8) | null |
| dob | | DATE | null |
| death | | VARCHAR2(1) | null |
| sex | | VARCHAR2(8) | null |
| mstatus | | VARCHAR2(8) | null |
| ssn | | VARCHAR2(16) | null |
| race | | VARCHAR2(8) | null |
| ethnic | | VARCHAR2(8) | null |
| religion | | VARCHAR2(8) | null |
| language | | VARCHAR2(8) | null |
| spouse_name | | VARCHAR2(100) | null |
| mother_name | | VARCHAR2(100) | null |
| mother_mn | | VARCHAR2(40) | null |
| father_name | | VARCHAR2(100) | null |
| maiden | | VARCHAR2(40) | null |
| pob_city | | VARCHAR2(30) | null |
| pob_state | | VARCHAR2(10) | null |
| pob_country | | VARCHAR2(20) | null |
| vip_flag | | VARCHAR2(8) | null |
| vet_status | | VARCHAR2(8) | null |
| status | | VARCHAR2(8) | null |
| fname_phonetic_code | | VARCHAR2(8) | null |
| lname_phonetic_code | | VARCHAR2(8) | null |
| mname_phonetic_code | | VARCHAR2(8) | null |
| mother_mn_phonetic_code | | VARCHAR2(8) | null |
| maiden_phonetic_code | | VARCHAR2(8) | null |
| spouse_phonetic_code | | VARCHAR2(8) | null |
| mother_phonetic_code | | VARCHAR2(8) | null |
| father_phonetic_code | | VARCHAR2(8) | null |
| drivers_license | | VARCHAR2(20) | null |
| drivers_license_st | | VARCHAR2(10) | null |
| dod | | DATE | null |
| death_certificate | | VARCHAR2(10) | null |
| nationality | | VARCHAR2(8) | null |
| citizenship | | VARCHAR2(8) | null |
| pension_no | | VARCHAR2(15) | null |
| pension_exp_date | | DATE | null |
| repatriation_no | | VARCHAR2(16) | null |
| district_of_residence | | VARCHAR2(8) | null |
| lga_code | | VARCHAR2(4) | null |
| military_branch | | VARCHAR2(4) | null |
| military_rank | | VARCHAR2(4) | null |
| military_status | | VARCHAR2(4) | null |
| dummy_date | | DATE | null |
| class1 | | VARCHAR2(20) | null |
| class2 | | VARCHAR2(20) | null |
| class3 | | VARCHAR2(20) | null |
| class4 | | VARCHAR2(20) | null |
| class5 | | VARCHAR2(20) | null |
| string1 | | VARCHAR2(40) | null |
| string2 | | VARCHAR2(40) | null |
| string3 | | VARCHAR2(40) | null |
| string4 | | VARCHAR2(40) | null |
| string5 | | VARCHAR2(40) | null |
| string6 | | VARCHAR2(40) | null |
| string7 | | VARCHAR2(100) | null |
| string8 | | VARCHAR2(100) | null |
| string9 | | VARCHAR2(100) | null |
| string10 | | VARCHAR2(255) | null |
| date1 | | DATE | null |
| date2 | | DATE | null |
| date3 | | DATE | null |
| date4 | | DATE | null |
| date5 | | DATE | null |
| transaction_no | <fk> | NUMBER(10) | null |

u_id = u_id

**ui_mrg_trans**

| | | | |
|---|---|---|---|
| ui_mrg_trans_id | <pk> | NUMBER(10) | not null |
| u_id | <fk> | VARCHAR2(15) | not null |
| u_id_from | <fk> | VARCHAR2(15) | not null |
| trans_no | <fk> | NUMBER(10) | not null |
| trans_no_from | <fk> | NUMBER(10) | null |
| unmrg_trans_no | <fk> | NUMBER(10) | null |
| unmrg_trans_no_from | <fk> | NUMBER(10) | null |

u_id = u_id
u_id = u_id_from

FROM UI_PERSON_HISTORY

transaction_no = transaction_no

transaction_no = transaction_no

transaction_no = transaction_no

transaction_no = transaction_no

transaction_no = transaction_no

transaction_no = transaction_no

transaction_no = transaction_no

transaction_no = transaction_no

**ui_transaction**

| transaction_no | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| u_id | | VARCHAR2(15) | not null |
| system | | VARCHAR2(8) | null |
| dept | | VARCHAR2(8) | null |
| terminal_id | | VARCHAR2(25) | null |
| session_id | | VARCHAR2(15) | null |
| update_facility_id | | VARCHAR2(20) | not null |
| update_function | | VARCHAR2(8) | not null |
| update_date | | DATE | not null |
| update_userid | | VARCHAR2(20) | not null |

u_id = u_id

transaction_no = transaction_no

**ui_phone**

| phone_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| u_id | <fk> | VARCHAR2(15) | not null |
| phone_type | | VARCHAR2(8) | not null |
| phone | | VARCHAR2(20) | not null |
| phone_ext | | VARCHAR2(6) | null |
| transaction_no | <fk> | NUMBER(10) | null |

**ui_person**

| u_id | <pk> | VARCHAR2(15) | not null |
|---|---|---|---|
| person_cat_code | | VARCHAR2(8) | null |
| last_name | | VARCHAR2(40) | not null |
| first_name | | VARCHAR2(40) | not null |
| middle_name | | VARCHAR2(30) | null |
| suffix | | VARCHAR2(10) | null |
| title | | VARCHAR2(8) | null |
| dob | | DATE | null |
| death | | VARCHAR2(1) | null |
| sex | | VARCHAR2(8) | null |
| mstatus | | VARCHAR2(8) | null |
| ssn | | VARCHAR2(16) | null |
| race | | VARCHAR2(8) | null |
| ethnic | | VARCHAR2(8) | null |
| religion | | VARCHAR2(8) | null |
| language | | VARCHAR2(8) | null |
| spouse_name | | VARCHAR2(100) | null |
| mother_name | | VARCHAR2(100) | null |
| mother_mn | | VARCHAR2(40) | null |
| father_name | | VARCHAR2(100) | null |
| maiden | | VARCHAR2(40) | null |
| pob_city | | VARCHAR2(30) | null |
| pob_state | | VARCHAR2(10) | null |
| pob_country | | VARCHAR2(20) | null |
| vip_flag | | VARCHAR2(8) | null |
| vet_status | | VARCHAR2(8) | null |
| status | | VARCHAR2(8) | null |
| fname_phonetic_code | | VARCHAR2(8) | null |
| lname_phonetic_code | | VARCHAR2(8) | null |
| mname_phonetic_code | | VARCHAR2(8) | null |
| mother_mn_phonetic_code | | VARCHAR2(8) | null |
| maiden_phonetic_code | | VARCHAR2(8) | null |
| spouse_phonetic_code | | VARCHAR2(8) | null |
| mother_phonetic_code | | VARCHAR2(8) | null |
| father_phonetic_code | | VARCHAR2(8) | null |
| drivers_license | | VARCHAR2(20) | null |
| drivers_license_st | | VARCHAR2(10) | null |
| dod | | DATE | null |
| death_certificate | | VARCHAR2(10) | null |
| nationality | | VARCHAR2(8) | null |
| citizenship | | VARCHAR2(8) | null |
| pension_no | | VARCHAR2(15) | null |
| pension_exp_date | | DATE | null |
| repatriation_no | | VARCHAR2(16) | null |
| district_of_residence | | VARCHAR2(8) | null |
| lga_code | | VARCHAR2(4) | null |
| military_branch | | VARCHAR2(4) | null |
| military_rank | | VARCHAR2(4) | null |
| military_status | | VARCHAR2(4) | null |
| dummy_date | | DATE | null |
| class1 | | VARCHAR2(20) | null |
| class2 | | VARCHAR2(20) | null |
| class3 | | VARCHAR2(20) | null |
| class4 | | VARCHAR2(20) | null |
| class5 | | VARCHAR2(20) | null |
| string1 | | VARCHAR2(40) | null |
| string2 | | VARCHAR2(40) | null |
| string3 | | VARCHAR2(40) | null |
| string4 | | VARCHAR2(40) | null |
| string5 | | VARCHAR2(40) | null |
| string6 | | VARCHAR2(40) | null |
| string7 | | VARCHAR2(100) | null |
| string8 | | VARCHAR2(100) | null |
| string9 | | VARCHAR2(100) | null |
| string10 | | VARCHAR2(255) | null |
| date1 | | DATE | null |
| date2 | | DATE | null |
| date3 | | DATE | null |
| date4 | | DATE | null |
| date5 | | DATE | null |
| transaction_no | <fk> | NUMBER(10) | null |

u_id = u_id

**ui_phone_history**

| phone_id | | NUMBER(10) | not null |
|---|---|---|---|
| u_id | <fk> | VARCHAR2(15) | not null |
| phone_type | | VARCHAR2(8) | null |
| phone | | VARCHAR2(20) | null |
| phone_ext | | VARCHAR2(6) | null |
| transaction_no | <fk> | NUMBER(10) | not null |

u_id = duplic_id

u_id = existing_id

**ui_duplic**

| ui_duplic_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| duplic_id | <fk> | VARCHAR2(15) | not null |
| existing_id | <fk> | VARCHAR2(15) | not null |
| weight | | FLOAT | not null |
| description | | VARCHAR2(30) | null |
| dummy_date | | DATE | null |
| resolved | | VARCHAR2(8) | null |
| transaction_no | <fk> | NUMBER(10) | null |

u_id = u_id

**ui_local_id**

| ul_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| u_id | <fk> | VARCHAR2(15) | not null |
| facility | <fk> | VARCHAR2(20) | not null |
| local_id | | VARCHAR2(25) | not null |
| status | | VARCHAR2(8) | null |
| transaction_no | <fk> | NUMBER(10) | null |

facility_code = facility

facility_code = facility

**ui_facility**

| facility_code | <pk> | VARCHAR2(20) | not null |
|---|---|---|---|
| description | | VARCHAR2(30) | null |
| status | | VARCHAR2(8) | null |
| addr1 | | VARCHAR2(30) | null |
| addr2 | | VARCHAR2(30) | null |
| city | | VARCHAR2(30) | null |
| state | | VARCHAR2(10) | null |
| zip | | VARCHAR2(8) | null |
| zipext | | VARCHAR2(4) | null |
| county | | VARCHAR2(20) | null |
| country | | VARCHAR2(20) | null |
| region_code | | VARCHAR2(8) | null |
| date_time | | DATE | null |
| id_length | | NUMBER | null |
| mask | | CHAR(1) | null |
| local_id_format | | VARCHAR2(35) | null |
| facility_mask | | CHAR(1) | null |
| next_local_id | | VARCHAR2(25) | null |
| allow_less_id_length | | VARCHAR2(1) | null |

**ui_local_id_history**

| ul_id | | NUMBER(10) | not null |
|---|---|---|---|
| u_id | <fk> | VARCHAR2(15) | not null |
| facility | <fk> | VARCHAR2(20) | null |
| local_id | | VARCHAR2(25) | null |
| status | | VARCHAR2(8) | null |
| transaction_no | <fk> | NUMBER(10) | not null |

u_id = u_id

**ui_alias**

| ui_alias_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| u_id | <fk> | VARCHAR2(15) | not null |
| last_name | | VARCHAR2(40) | not null |
| first_name | | VARCHAR2(40) | not null |
| middle_name | | VARCHAR2(30) | null |
| lname_phonetic_code | | VARCHAR2(8) | null |
| fname_phonetic_code | | VARCHAR2(8) | null |
| mname_phonetic_code | | VARCHAR2(8) | null |
| transaction_no | <fk> | NUMBER(10) | null |

u_id = u_id

**ui_alias_history**

| ui_alias_id | | NUMBER(10) | not null |
|---|---|---|---|
| u_id | <fk> | VARCHAR2(15) | not null |
| last_name | | VARCHAR2(40) | null |
| first_name | | VARCHAR2(40) | null |
| middle_name | | VARCHAR2(30) | null |
| lname_phonetic_code | | VARCHAR2(8) | null |
| fname_phonetic_code | | VARCHAR2(8) | null |
| mname_phonetic_code | | VARCHAR2(8) | null |
| transaction_no | <fk> | NUMBER(10) | not null |

### ui_ctrl_rule

| | | | |
|---|---|---|---|
| ui_ctrl_rule_id | <pk> | NUMBER(10) | not null |
| rule_name | | VARCHAR2(16) | not null |
| root_file | | VARCHAR2(16) | not null |
| read_only | | CHAR(1) | not null |
| in_use | | CHAR(1) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |
| update_date | | DATE | null |
| update_userid | | VARCHAR2(20) | null |

ui_ctrl_rule_id = ui_ctrl_rule_id

### ui_ctrl_file

| | | | |
|---|---|---|---|
| ui_ctrl_file_id | <pk> | NUMBER(10) | not null |
| ui_ctrl_rule_id | <fk> | NUMBER(10) | not null |
| file_type | | VARCHAR2(3) | not null |
| file_name | | VARCHAR2(18) | not null |
| file_ext | | VARCHAR2(3) | not null |
| file_content | | LONG | null |
| content_date | | DATE | not null |
| last_synch_date | | DATE | null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |
| update_date | | DATE | null |
| update_userid | | VARCHAR2(20) | null |

ctrl_file_id = ui_ctrl

### ui_ctrl_table

| | | | |
|---|---|---|---|
| ui_ctrl_table_id | <pk> | NUMBER(10) | not null |
| table_name | | VARCHAR2(30) | not null |
| description | | VARCHAR2(48) | null |
| read_only | | CHAR(1) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |
| update_date | | DATE | null |
| update_userid | | VARCHAR2(20) | null |

ui_ctrl_table_id = ui_ctrl_table_id

### ui_ctrl_column

| | | | |
|---|---|---|---|
| ui_ctrl_column_id | <pk> | NUMBER(10) | not null |
| ui_ctrl_table_id | <fk> | NUMBER(10) | not null |
| column_name | | VARCHAR2(30) | not null |
| description | | VARCHAR2(48) | null |
| read_only | | CHAR(1) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |
| update_date | | DATE | null |
| update_userid | | VARCHAR2(20) | null |

### ui_misc_opt_control

| | | | |
|---|---|---|---|
| control_type | <pk> | varchar2(8) | not null |
| description | | varchar2(40) | not null |

control_type = control_type

### ui_misc_option

| | | | |
|---|---|---|---|
| misc_option_id | <pk> | number(10) | not null |
| country_code | <fk> | varchar2(8) | not null |
| control_type | <fk> | varchar2(8) | not null |
| option_name | | varchar2(40) | not null |
| value | | varchar2(40) | not null |

country_code = country_code

### ui_misc_opt_country

| | | | |
|---|---|---|---|
| country_code | <pk> | varchar2(8) | not null |
| country_name | | varchar2(40) | null |

### ui_person_x_name

| | | | |
|---|---|---|---|
| u_id | <pk> | varchar2(15) | not null |
| x_last_name | | varchar2(40) | not null |
| x_first_name | | varchar2(40) | not null |
| x_mother_mn | | varchar2(40) | null |
| x_maiden | | varchar2(40) | null |
| x_middle_name | | varchar2(30) | null |

### ui_nickname

| | | | |
|---|---|---|---|
| ui_nickname_id | <pk> | NUMBER(10) | not null |
| formal_name | | VARCHAR2(40) | not null |
| nick_name | | VARCHAR2(40) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |
| update_date | | DATE | null |
| update_userid | | VARCHAR2(20) | null |

### ui_ctrl_file_hist

| | | | |
|---|---|---|---|
| ui_ctrl_file_hist_id | | NUMBER(10) | not null |
| ui_ctrl_file_id | <fk> | NUMBER(10) | not null |
| file_type | | VARCHAR2(3) | not null |
| file_name | | VARCHAR2(18) | not null |
| file_ext | | VARCHAR2(3) | not null |
| file_content | | LONG | null |
| content_date | | DATE | not null |
| save_date | | DATE | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |
| update_date | | DATE | null |
| update_userid | | VARCHAR2(20) | null |

### ui_ctrl_field

| | | | |
|---|---|---|---|
| ui_ctrl_field_id | <pk> | NUMBER(10) | not null |
| field_name | | VARCHAR2(2) | not null |
| field_type | | VARCHAR2(2) | not null |
| field_length | | NUMBER(10) | not null |
| field_missing | | VARCHAR2(2) | not null |
| description | | VARCHAR2(48) | null |
| ui_ctrl_column_id | | NUMBER(10) | null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |
| update_date | | DATE | null |
| update_userid | | VARCHAR2(20) | null |

### ui_table

| | | | |
|---|---|---|---|
| ui_table_id | <pk> | NUMBER(10) | not null |
| ui_table_name | | VARCHAR2(30) | not null |

ui_table_id = ui_table_id

### ui_table_column

| | | | |
|---|---|---|---|
| table_column_id | <pk> | NUMBER(10) | not null |
| ui_table_id | <fk> | NUMBER(10) | not null |
| column_name | | VARCHAR2(30) | not null |
| default_label | | VARCHAR2(40) | not null |
| label | | VARCHAR2(40) | not null |
| visible | | CHAR(1) | not null |
| required | | CHAR(1) | not null |
| read_only | | CHAR(1) | not null |

### control_sec

| | | | |
|---|---|---|---|
| ctrl_key | <pk> | VARCHAR2(10) | not null |
| description | | VARCHAR2(50) | null |
| ctrl_value | | VARCHAR2(10) | null |
| create_date | | DATE | null |

### ui_no_passwd

| | | | |
|---|---|---|---|
| no_passwd | <pk> | VARCHAR2(20) | not null |

🗀 pk_idx_ui_no_passwd

### ui_alias_x_name

| | | | |
|---|---|---|---|
| ui_alias_id | <pk> | number(10) | not null |
| u_id | | varchar2(15) | not null |
| x_last_name | | varchar2(40) | not null |
| x_first_name | | varchar2(40) | not null |
| x_middle_name | | varchar2(30) | null |

**stc_user**

| usersl_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| user_id | | VARCHAR2(20) | not null |
| last_name | | VARCHAR2(30) | not null |
| first_name | | VARCHAR2(30) | not null |
| middle_initial | | VARCHAR2(1) | null |
| user_desc | | VARCHAR2(100) | null |
| email_address | | VARCHAR2(100) | null |
| admin | | VARCHAR2(1) | not null |
| eff_date | | DATE | not null |
| exp_date | | DATE | null |
| change_pass | | VARCHAR2(1) | not null |
| login_attempts | | NUMBER(10) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

pk_stc_user2

usersl_id = usersl_id

**ui_login_current**

| ui_login_current_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| usersl_id | <fk> | NUMBER(10) | not null |
| login_time | | DATE | not null |
| logout_time | | DATE | null |
| update_time | | DATE | null |
| session_id | | VARCHAR2(15) | null |

usersl_id = usersl_id

**ui_notify_user**

| ui_notify_user_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| event_code | | VARCHAR2(5) | not null |
| usersl_id | <fk> | NUMBER(10) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

**stc_user_region**

| user_region_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| usersl_id | <fk> | NUMBER(10) | not null |
| region_code | | VARCHAR2(8) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

usersl_id = usersl_id

**ui_login**

| ui_login_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| usersl_id | <fk> | NUMBER(10) | not null |
| login_time | | DATE | not null |
| logout_time | | DATE | null |
| update_time | | DATE | null |
| session_id | | VARCHAR2(15) | null |

usersl_id = usersl_id

**ui_user_passwd_hist**

| ui_passwd_hist_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| usersl_id | <fk> | NUMBER(10) | not null |
| passwd | | VARCHAR2(20) | not null |
| use_date | | DATE | not null |

ui_user_passwd_hist_1

usersl_id = usersl_id

**stc_user_acc**

| user_acc_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| usersl_id | <fk> | NUMBER(10) | not null |
| acc_def_id | <fk> | NUMBER(10) | not null |
| eff_date | | DATE | not null |
| exp_date | | DATE | null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

acc_def_id = acc_def_id

**stc_common_detail**

| common_detail_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| common_header_id | <fk> | NUMBER(10) | not null |
| code | | VARCHAR2(20) | not null |
| descr | | VARCHAR2(50) | not null |
| read_only | | CHAR(1) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

common_header_id = common_header_id

**stc_common_header**

| common_header_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| appl_id | <fk> | NUMBER(10) | not null |
| code | | VARCHAR2(8) | not null |
| descr | | VARCHAR2(50) | not null |
| read_only | | CHAR(1) | not null |
| max_input_len | | NUMBER(10) | not null |
| typ_table_code | | VARCHAR2(3) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

usersl_id = usersl_id

**stc_user_group**

| usergp_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| group_id | <fk> | NUMBER(10) | not null |
| usersl_id | <fk> | NUMBER(10) | not null |
| eff_date | | DATE | not null |
| exp_date | | DATE | null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

group_id = group_id

**stc_group**

| group_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| group_name | | VARCHAR2(30) | not null |
| group_desc | | VARCHAR2(100) | null |
| eff_date | | DATE | not null |
| exp_date | | DATE | null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

group_id = group_id

**stc_group_acc**

| group_acc_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| group_id | <fk> | NUMBER(10) | not null |
| acc_def_id | <fk> | NUMBER(10) | not null |
| eff_date | | DATE | not null |
| exp_date | | DATE | null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

acc_def_id = acc_def_id

**stc_acc_def**

| acc_def_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| module_id | <fk> | NUMBER(10) | not null |
| code | | VARCHAR2(5) | not null |
| descr | | VARCHAR2(50) | null |
| create_date | | DATE | null |
| create_userid | | VARCHAR2(15) | null |

module_id = module_id

**stc_module**

| module_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| appl_id | <fk> | NUMBER(10) | not null |
| code | | VARCHAR2(5) | not null |
| descr | | VARCHAR2(30) | not null |
| read_only | | CHAR(1) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

appl_id = appl_id

**stc_appl**

| appl_id | <pk> | NUMBER(10) | not null |
|---|---|---|---|
| code | | VARCHAR2(8) | not null |
| descr | | VARCHAR2(30) | not null |
| read_only | | CHAR(1) | not null |
| create_date | | DATE | not null |
| create_userid | | VARCHAR2(20) | not null |

appl_id = appl_id

# Upgrading the e*Index Schema Files

## About this Chapter

### Overview

This chapter presents the background information and the step-by-step instructions you need to upgrade the e*Index Schema files from version 4.1.1 to version 4.5.2.

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 Upgrade** | Learn about the Schema upgrade process and the files in the Schema |
| **Install the Upgrade Files** | Learn how to install the upgrade files and about the files you may need to customize |

# What's Inside

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

# Learning About the Upgrade Process

## Overview

This section of the chapter provides background information about upgrading the e*Index Schema files for e*Gate, and the Schema files that are installed for e*Index.

## Getting Started

Installing e*Index version 4.5.2 may require some modifications to your e*Gate and Oracle e*Way environments. We highly recommend that a separate e*Gate and Oracle 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 4.5.0 or later

- The Oracle e*Way 4.5.0 or later

- HL7 Templates 4.5.0 or later (only if you process HL7 messages)

- e-Mail e*Way 4.5.0 or later (only if you will use the Event Notification function of e*Index Security)

- e*Index 4.5.2

You also need Oracle Client version 8.1.7 for database connectivity.

## Platform-specific Information

The upgrade procedures for Windows NT and UNIX differ slightly, but both installations begin on a machine running Windows 2000 or Windows NT 4.0 with SP4 or later. The extensions for the dynamic link library files that are installed during this process may vary depending on the platform you are running. This guide refers to these files by the Windows extension, **.dll**. The variable **<eGate>** is used to specify your e*Gate environment on any platform.

## About the e*Index Schema

When you install the e*Index Schema upgrade files, you can install either the Schema components or the Schema components plus a sample Schema. Typically, you can upgrade the e*Index Schema by simply installing the Schema components, but for this upgrade, you need to install both the Schema components and the sample Schema due to the number of

differences between versions 4.1.1 and 4.5.2.   Install the new Schema on a test server, away from the production environment.  You can customize the sample Schema for use in production using your e*Index 4.1.1 Schema as a reference.  For a complete description of the e*Index Schema files, see "Learning About e*Index Schema Components" in Chapter 3 of the *e*Index Global Identifier Technical Reference* .

## Schema Components

The Schema components files are installed in the registry of the e*Gate server and in the e*Gate client.  On the server, the files are installed in **\eGate\Server\Registry\repository\default**.  Generally, when the e*Index Schema is upgraded, only the component files change.  However, for this release the sample Schema files were changed as well.  The Schema component files are generally binary files that cannot be customized, with the exception of two customizable files:  **ui-fns.monk** and **eiEvent.ssc**.  Refer to the diagrams on beginning on page 2-15 for the locations of these files.

## Sample Schema

The sample Schema files are installed in the registry of the e*Gate server and in the e*Gate client.  On the server, the sample Schema files are installed in **\eGate\Server\Registry\repository\***schema_name***\runtime**, where *schema_name* is the name you specify when you perform the installation.  You can customize any of the sample schema files.  Refer to the diagrams beginning on page 1-15 for the locations of these files.

Installing the sample Schema creates five e*Ways and their associated configuration files.  The e*Ways include the sending e*Way, the polling e*Way, and three file e*Ways for testing the program.  The sample Schema also includes the sample Collaboration script, **uidb.dsc**, used by the sending e*Way to process incoming data.  You need to modify this script for use in your production environment.  The sample Schema also provides a sample Monk file, **ui-process-person.monk**, which is called by **uidb.dsc** and defines how records are initially processed.  **ui-process-person.monk** replaces the old Monk API **alta-process-person**, and is a configurable version of the API.  The sample Schema also includes Monk files that define database connectivity and create the Monk lists used as parameters for the Monk APIs.

# Performing the Upgrade

## Overview

To install the Schema for e\*Index, 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 Oracle e\*Way
- Step 4: Install or Upgrade the e-Mail e\*Way (optional)
- Step 5: Install or Upgrade the Oracle Client Software
- Step 6: Install the e\*Index Schema Files
- Step 7: Verify the e\*Index Schema File Structure
- Step 8: Customize the e\*Index 4.5.2 Schema
- Step 9: Test and Move to Production

## Step 1: Back up the e\*Gate Environment

It is important to make a FULL backup of the environment for safekeeping before making any changes to your e\*Gate environment.

## Step 2: Install or Upgrade e\*Gate

Before you install the e\*Index 4.5.2 Schema, you need to either install or upgrade to e\*Gate 4.5.0 or later.  Follow the instructions for installing or upgrading e\*Gate in the *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 4.5.0 or later, install it on a separate environment for testing purposes, away from your production e\*Gate environment.

## Step 3: Install or Upgrade the Oracle e\*Way

Before you can work with the e\*Index 4.5.2 Schema, you need to install or upgrade to the e\*Way Intelligent Adapter for Oracle (Oracle e\*Way) version 4.5.0 or later.  The Oracle e\*Way must reside on the e\*Gate server on which you install the e\*Index Schema.  For information on installing or upgrading the Oracle e\*Way, refer to the e\**Way Intelligent Adapter for Oracle User's Guide*.

## Step 4: Install or Upgrade the e-Mail e\*Way (optional)

You need to install or upgrade the e-Mail e\*Way Intelligent Adapter (e-Mail e\*Way) only if you will be using the Event Notification function in e\*Index

Security (for more information, see chapter 3 of the *e\*Index Security User's Guide* for version 4.5.2).  For information on installing, upgrading, and implementing the e-Mail e\*Way, refer to the *e-Mail e\*Way Intelligent Adapter User's Guide*.

# Step 5: Install or Upgrade the Oracle Client Software

In order for the e\*Index e\*Way to connect with the e\*Index database, you need to install or upgrade to Oracle Client 8.1.7 on the e\*Gate host on which the e\*Index Schema is installed.  This may be the Registry Host or a Participating Host.  Before you install the client software, you should make sure that no previous versions of the software are installed.  Make sure to include the Oracle network administration and application development tools in your installation.  After you install Oracle, modify **tnsnames.ora** by adding a stanza for the new e\*Index 4.5.2 database (for more information, see "Step 6: Modify **tnsnames.ora**" in Chapter 4 of this guide).

For information about installing Oracle Client, refer to the *Oracle8i Installation Guide*.

# Step 6: Install the e\*Index Schema Files

To begin the installation process, insert the e\*Index 4.5.2 installation CD-ROM into the CD-ROM drive on your computer, and make sure no other Windows applications are running.

### ▶ To install the e\*Gate interface files

Before you begin:

✓ Complete "Step 1: Back up the e\*Gate Environment" through "Step 5: Install or Upgrade the Oracle Client Software"

✓ Make sure that all Windows applications are closed

✓ Make sure you initially install the Schema on a test server to make sure no files for the production 4.1.1 Schema are overwritten.  When you are ready to move to production with your 4.5.2 Schema, you can transfer it to the production server.

**1**  Insert the e\*Index installation CD-ROM into the CD-ROM drive of your computer

**2**  If Autorun is enabled, the setup program automatically starts.  Otherwise:

- On the Windows desktop, double-click the **My Computer** icon and then open the CD-ROM directory.

• Double-click the file name **Setup.exe** to initiate the process that installs the e*Index Schema files. The Welcome window appears, reminding you to close all Windows programs.

e*Index is installed using a standard InstallShield Wizard
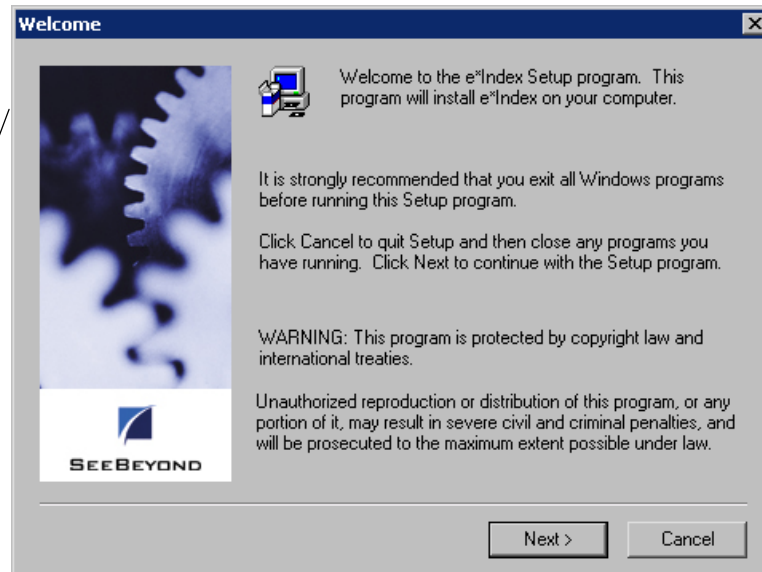


**3**  Do one of the following:

*To close any open Windows programs*, click **Cancel**, close the programs, and then repeat step 2.

Cancel button

*To continue with the installation process without closing any external programs*, click **Next**. The Software License Agreement window appears.

Next button

You must agree to the terms of the license agreement in order to proceed

**Yes**

Yes button
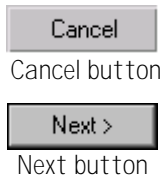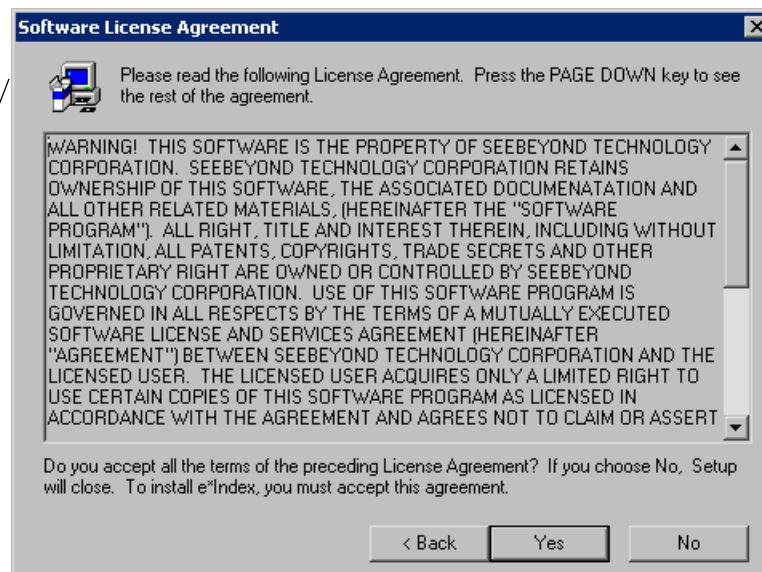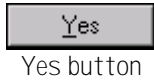
**4**   If you agree to the license agreement, click **Yes**.  The Information window appears with a list of requirements to remind you of the applications you need to install before installing e*Index.

The Information window reminds you to have Windows NT, Internet Explorer, and database software installed

**Next >**

Next button
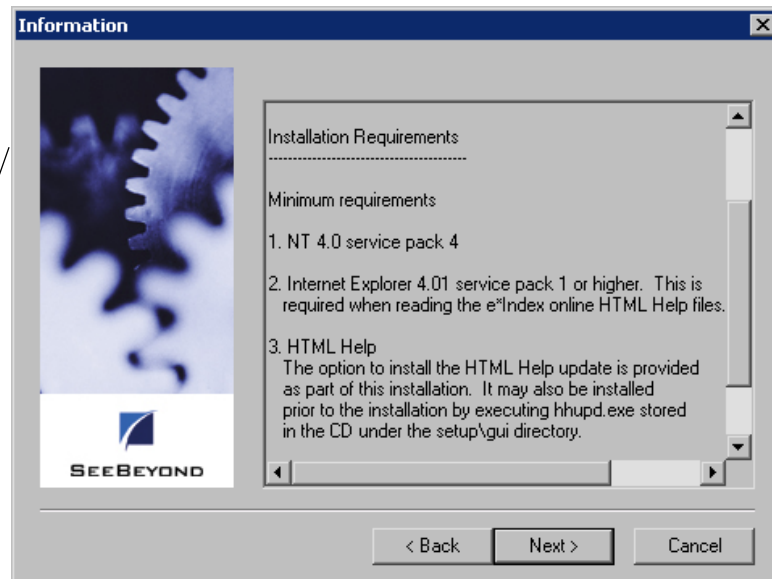
**5**   Click **Next**.  The User Information window appears.

Enter your name and your company name on the  User Information window

**Next >**

Next button

**6**   In the **Name** and **Company** fields, enter your name and your company's name, and then click **Next**.  The Choose Destination Location window appears.

The Choose
Destination
Location window
displays the
e*Gate client path
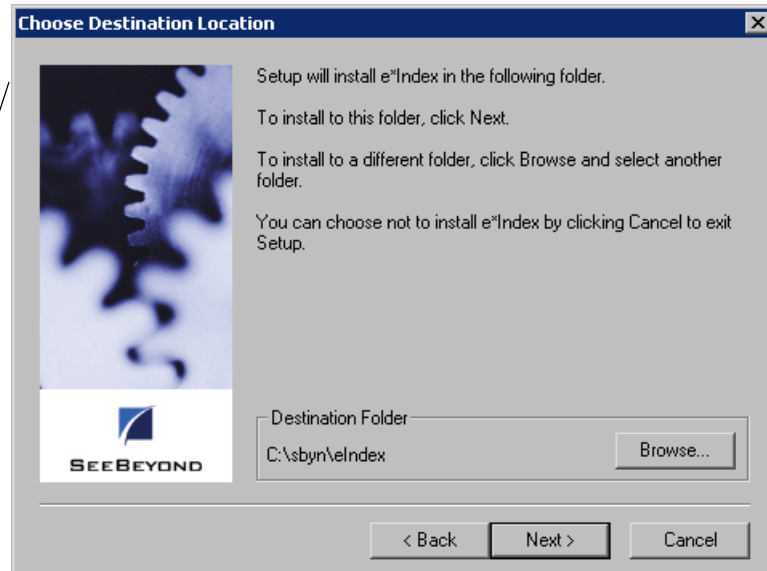
**7** Do one of the following:

*To accept the default folder that appears in the Destination Folder path*, click
   **Next**.

*To change the location in which the files will be installed*:

- Click **Browse**.

- In the Choose Folder dialog, type or select the path where you
  want to install the files.

- Click **OK**. The new path you specified appears in the
  Destination Folder path.

Specify a new
path in the
Choose Folder
dialog
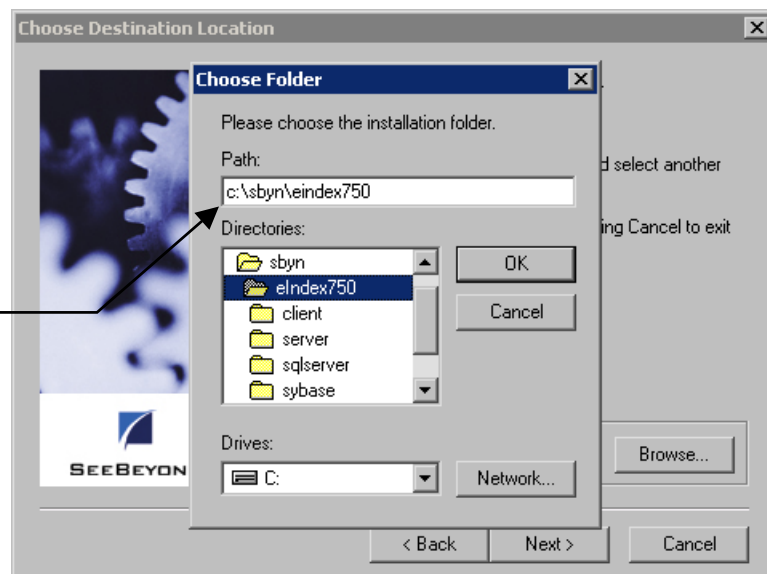
*Note:*   *If the path you specified does not exist, a dialog appears asking if you
want to have the folder created.  If you select **Yes**, Setup creates the specified path
for you.*

Next >

*Next button*

**8** After you specify the installation path, click **Next** on the Choose Destination Location taskbar. The Select Components window appears.

Select the components you want to install on the Select Components window
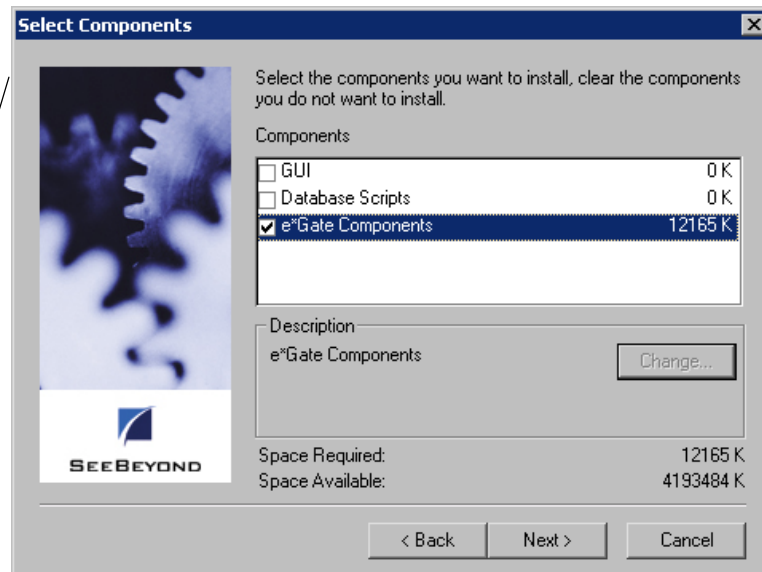
**9** In the Components box, select **e*Gate Components**, and make sure that no other components are selected.

*Note: At the bottom of the window, the required space and the available space appears so you can see how much space each component requires compared to the space you have available on your machine.*

Next >

*Next button*

**10** Click **Next**. The Select e*Gate Platforms window appears.

You can specify the platforms on which your e*Gate host servers are running
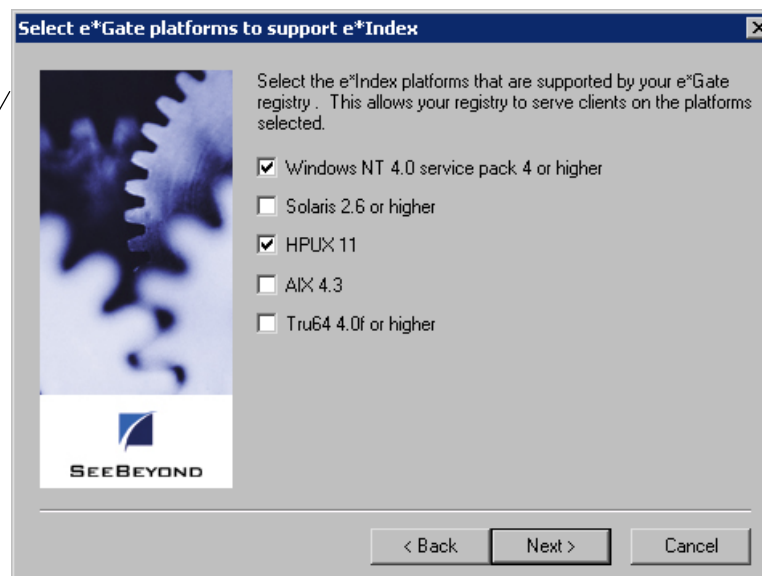
| Next > |
| :---: |
| Next button |

**11** Select the platforms on which you will be running the e*Index Schemas, and then click **Next**. The Registry Hostname and Schema window appears.

On the Registry Hostname and Schema window, specify information about your e*Gate environment



| Next > |
| :---: |
| Next button |

**12** In the **Registry** field, enter the name of your e*Gate registry host, and then click **Next**. The e*Index Demo Schema window appears
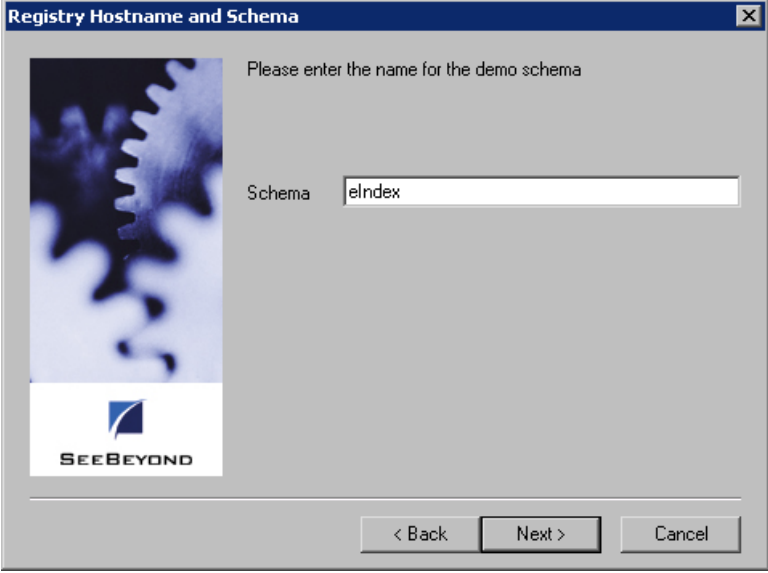
On the e*Index Demo Schema, specify that you want to install a demo Schema



| Next > |
| :---: |
| Next button |

**13** Select **Yes** and then click **Next**. The Registry Hostname and Schema window appears. Continue to step 13.

On the Registry Hostname and Schema window, specify a name for the sample Schema
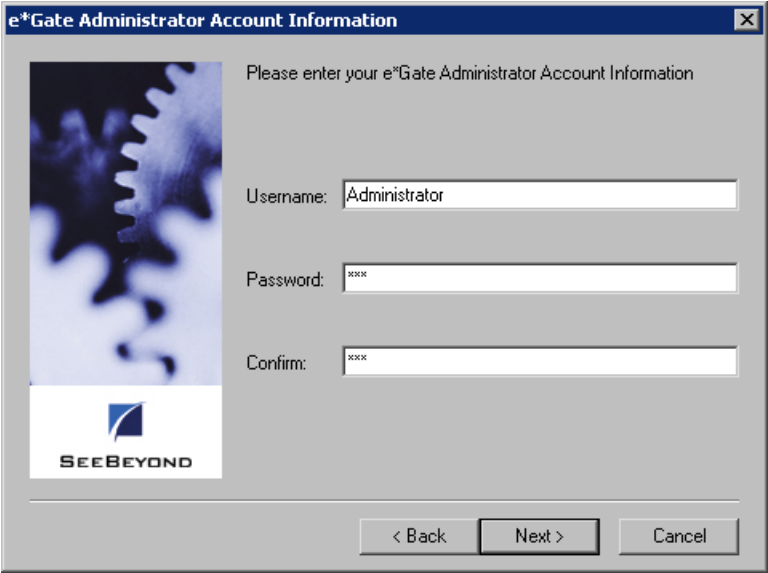
Next button

**14** In the **Schema** field, enter a name for the sample Schema.

> *Important!*  *Be sure to use a unique Schema name.  If you enter the name of an existing e*Index Schema, the new Schema overwrites the existing Schema files.*

Next button

**15** Click **Next**.  The e*Gate Administrator Account Information window appears.

On the e*Gate Administrator Account Information window, specify information about your e*Gate administrator account

**16** In the **Username** and **Password** fields, enter the administrator user ID and password for your e*Gate system.  Re-enter the password in the **Confirm** field.

Next >

*Next button*

**17** Click **Next**.  The Select Program Folder window appears.



Select the
program folder
to which you
want to add the
e*Index icons
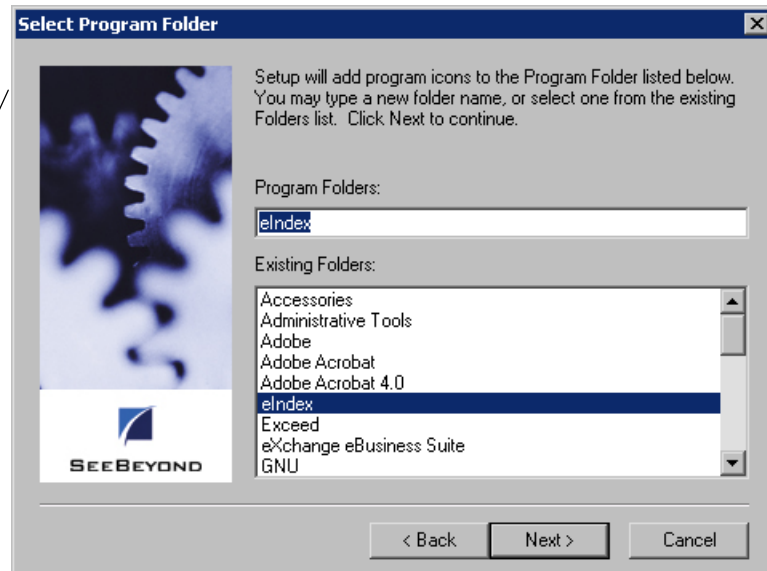
Next >

*Next button*

**18** In the **Program Folders** field, enter the name of the program folder to which you want to add the e*Index icons, and then click **Next**.  The Check Setup Information window appears.

On the Check
Setup Information
window, verify
the components
you are installing
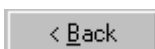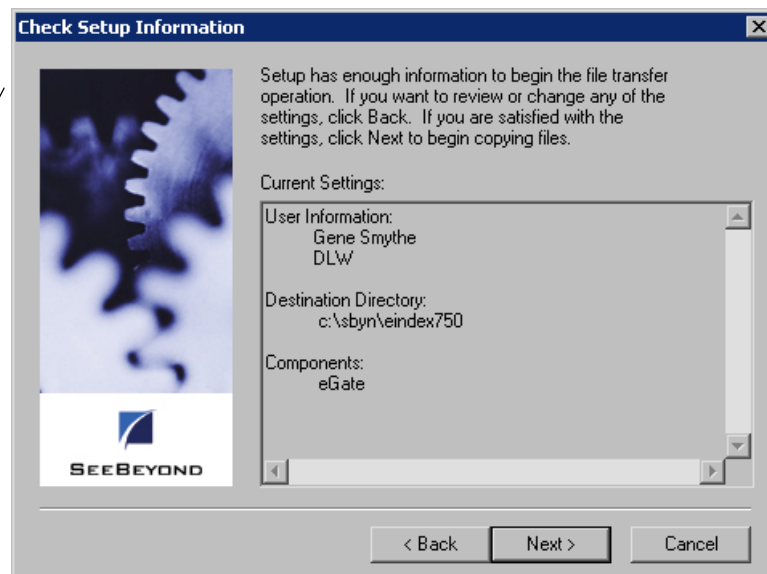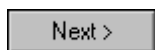


**19** Verify the information you specified, and do one of the following:

< Back

*Back button*

*To change any of the options you selected*, click **Back**, and make the necessary changes.

Next >

*Next button*

*To install the files in the specified directory*, click **Next**.  The Setup Complete window appears after the files are installed and committed to the e*Gate registry.

Finish

*Finish button*

**20** The Setup Complete window indicates that all necessary files are installed. Click **Finish** to conclude the setup process.

The Setup Complete window indicates that the files have been installed

**Setup Complete**

Setup has finished installing e*Index on your computer.

Setup can launch the Read Me file and e*Index. Choose the options you want below.

Click Finish to complete Setup.

SEEBEYOND

< Back    Finish
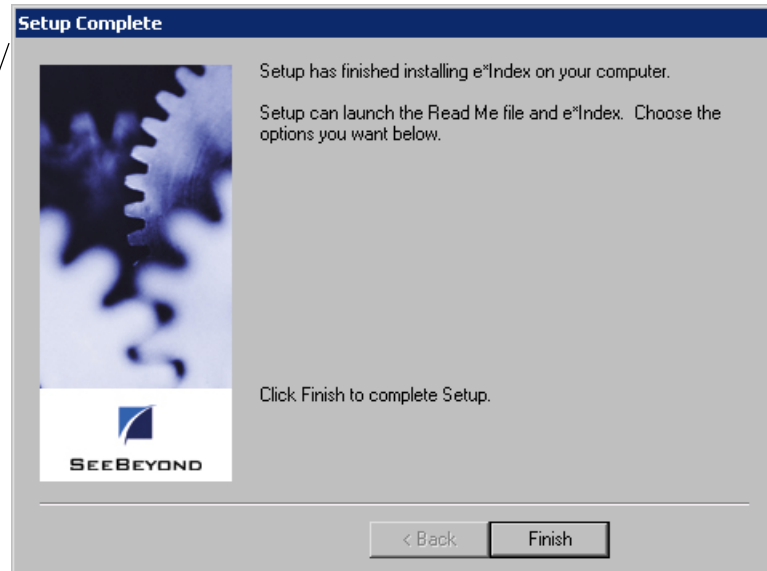
**21** Continue to "Step 7: Verify the e*Index Schema File Structure."

# Step 7: Verify the e\*Index Schema File Structure

After you install the e\*Index Schema files, your file structure on the e\*Gate server should resemble the illustration below. The variable *schema_name* refers to the name you assigned the new Schema when you installed the files.

```
<eGate>/server/           default/─────── .lockdir/
registry/repository                       ui.ctl
                                          bin/──────────── stc_uimonkext.ctl

                                                           .lockdir/
                                                           codesets/──────── .lockdir/
                                                                             codesets.map
                                                                             IBM-437.cex

                                                           <OS>/──────────── .lockdir/
                                                                             RealtimeC.dll (or
                                                                             libRealTimC.so for TRU64)
                                                                             stc_ui.dll
                                                                             stc_uimonkext.dll

                          monk_library/Ui/──────────────── .lockdir/
                                                           ui-fns.monk
                                                           ui-process-person.monk

                          monk_scripts/Ui/──────────────── .lockdir/
                                                           eiEvent.ssc


                                                           stcewfile/───────── .lockdir/
                                                                               ewUIEATER.cfg
                                                                               ewUIEATER.sc
                                                                               ewUIFEEDER.cfg
                                          configs/────────                     ewUIFEEDER.sc
                                                                               ewUIPOLLEATER.cfg
                                                                               ewUIPOLLEATER.sc

                                                           stcewgenericmonk/── .lockdir/
                                                                               ewUIDB.cfg
                                                                               ewUIDB.sc
                                                                               ewUIPOLL.cfg
                                                                               ewUIPOLL.sc

        <schema>/runtime/─
                                          monk_library/Ui/── .lockdir/
                                                             ui-custom.monk
                                                             ui-stdver-eway-funcs.monk

                                          monk_scripts/Ui/── .lockdir/
                                                             uidb.dsc.ssc
```

When you install the e*Index Schema files, the files are also copied to the installation path in **\client\eGate_schema\demo** and **\client\eGate_schema\default**. You can delete the Schema files from the installation path if you do not need them to create Schemas.

In addition, the Initial Load program for e*Index is installed in **\client\eGate_schema\init_load**. You can use these files when you load legacy data from your existing systems into the e*Index database. For more information, see your *e*Index Initial Load User's Guide* for version 4.5.2.

## Step 8: Customize the e*Index 4.5.2 Schema

Before you can use your e*Index Schema in a production environment, you need to customize the e*index 4.5.2 Schema files to process data in the same manner as your production 4.1.1 Schema. The following Schema files most likely need to be modified. Most of these modifications can be performed on the e*Gate editors or a standard text editor, and should be only performed by one who is familiar with e*Gate Schemas and processes. By default, the e*Index 4.5.2 Schema processes data in the same manner as the default 4.1.1 Schema, but the Event Type and ETD are different.

- uidb.dsc
- eiEvent.ssc (previously A0Xui.ssc in version 4.1.1)
- ewUIDB.cfg
- ewUIPOLL.cfg
- ui-custom.monk
- ui-fns.monk
- ui-process-person.monk
- ui-stdver-eway-funcs.monk

For information about the differences between the e*Index 4.1.1 and 4.5.2 Schemas and API calls, see "e*Index Schema Modifications" in the *e*Index 4.1.1 to 4.5.2 Release Bulletin*. Refer to chapters 3 and 4 of your *e*Index Global Identifier Technical Reference* for more information about customizing these files. Due to the differences in the ETDs between the two versions, you may need remap some of the data fields between external systems and e*Index.

## Step 9: Test and Move to Production

Once you have installed the new Schema files and made the necessary customizations, test the Schema to be sure it is processing data in the correct manner. Once the Schema is tested and approved, you can move it into production. Your new Schema should be ready for go-live at the time you need to queue daily feeds for the database migration (described in "Step 9: Begin Queuing 4.1.1 and 4.5.2 Daily Transactions" in chapter 5 of this guide).

# Upgrading the GUIs

## About this Chapter

### Overview

This chapter presents the background information and the step-by-step instructions you need to upgrade the e*Index GUIs and publications from version 4.1.1 to 4.5.2.

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 GUI and Publications | Learn about the GUI files and the e*Index Electronic Library |

| Upgrade the GUI and Publications | Learn how to upgrade the GUI and publications, and to make any required modifications to the GUI files |

# What's Inside

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

# Learning About the GUIs and Publications

## Overview

This section of the chapter provides background information about upgrading the e*Index GUI and the online publications on the Quality Workstation.

## Getting Started

In order to successfully install and use the e*Index GUI, the hardware and software items listed below are required.  Be sure your workstation meets these requirements before proceeding.

- ■ Client Hardware
    - ▤ Windows 2000 SP2, Windows NT 4.0 SP4 or higher, Windows 95, or Windows 98
    - ▤ Pentium 90 or higher
    - ▤ 32MB memory
    - ▤ 500MB disk space
    - ▤ VGA or higher
    - ▤ NIC cards
    - ▤ Valid TCP/IP addresses
    - ▤ Network connections
    - ▤ CD-ROM drive

- ■ Client Software
    - ▤ Oracle 8.1.7 Client (including Oracle 8.1.7 SQL *Net and Oracle 8.1.7 SQL Plus)
    - ▤ Internet Explorer 3.01 or higher (for online help)
    - ▤ e*Index 4.5.2

## GUI Upgrade Process

To perform the upgrade of the e*Index GUIs, you only need to uninstall or remove previous versions of the GUIs and install the new 4.5.2 version on each client workstation. Once you verify that your Oracle environment is configured correctly, and the e*Index initialization file correctly defines the e*Index databases and system, the GUIs are operational. You can also copy updated versions of the e*Index publications to the client workstations.

## How Do I View the Publications?

Once you copy the e*Index electronic library to your workstation, you can view the documents online using Adobe® Acrobat® Reader. You can also print the files to any postscript printer. If you do not have Acrobat Reader installed on your Workstation, you can install it from the Internet at **http://www.adobe.com**. For more information on the e*Index electronic library, read the **Readme.wri** file included with your publications.

The electronic library includes links between documents, a navigational document named **Welcome.pdf**, an index that allows you to search among all publications, and a feedback form for you to provide us with your comments. The library requires an additional 18MB of disk space.

# Performing the Upgrade

## Overview

To upgrade the e*Index GUI on your client workstations, you must complete the following steps:

- Step 1: Back up the GUI Files
- Step 2: Upgrade Oracle Client
- Step 3: Uninstall the Current e*Index GUI
- Step 4: Install the new GUI
- Step 5: Copy the Publications
- Step 6: Modify **tnsnames.ora**
- Step 7: Customize **stc_ua.ini**
- Step 8: Register the Online Help Support File
- Step 9: Reboot the Computer

## Step 1: Back up the GUI Files

Before installing the upgrade files, make a backup copy of your e*Index GUI environment for safekeeping.  After the installation, you may want to refer back to your initialization file, **stc_ua.ini**, to modify the new initialization file.

## Step 2: Upgrade Oracle Client

You may already have the correct version of Oracle Client installed on the client workstations for e*Index.  The required version of Oracle Client for e*Index 4.5.2 is 8.1.7.  If you have a previous version of Oracle, upgrade to the later version now.  For more information about upgrading Oracle Client, see the *Oracle8i Installation Guide*.

## Step 3: Uninstall the Current e*Index GUI

Once you back up **stc_ua.ini**, remove the existing e*Index GUI.  To uninstall the e*Index GUI, you can simply delete all files in the e*Index GUI home directory, or, if no other e*Index components are installed on the client machine, you can use the Add/Remove Programs function of the Control Panel to remove e*Index.

## Step 4: Install the GUI

Installing the 4.5.2 GUI is very similar to the process you followed to install the e*Gate schema files earlier in chapter 3 of this guide.

### ▶ To install the GUI files

Before you begin:

✓ Make sure that your workstation meets the requirements listed on page 4-**Error! Bookmark not defined.** of this chapter

✓ Complete "Step 1: Back up stc_ua.ini" through "Step 3: Uninstall the Current e*Index GUI"

**Change...**

*Change button*

**1** Follow steps 1 through 7 under "Step 6: Install the e*Index Schema Files" in chapter 3 of this guide.  The Select Components window should now be visible.

*Use the Select Components window to specify the components you want to install*
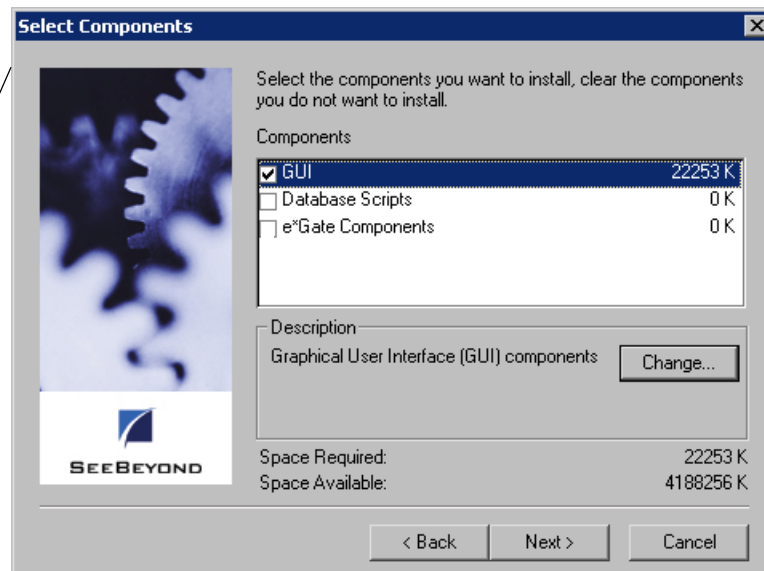


**2** On the Select Components window, select **GUI**.

**Change...**

*Change button*

**3** To verify which GUI components are being installed, highlight **GUI**, and then click **Change**.  The Select Sub-components window appears.

You can select
which GUI files to
install from the
Select Sub-
components
window
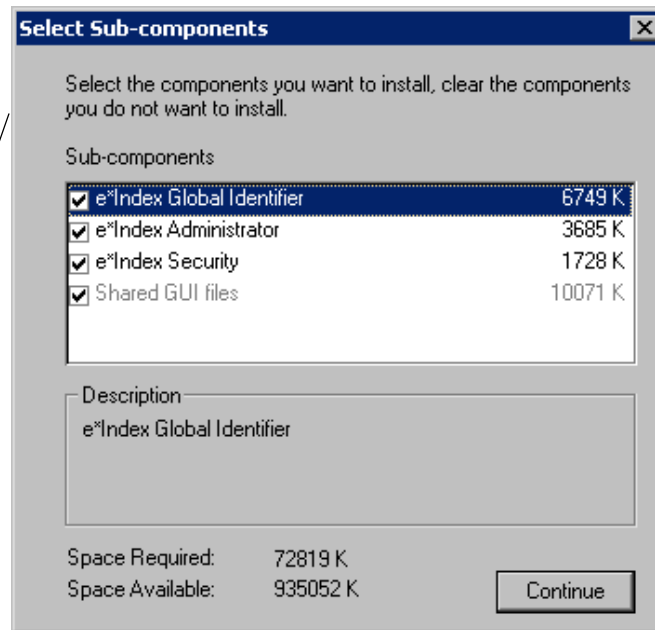
**4**   Select only the applications you want to install, and then click
**Continue**.

*Continue button*

---

*Note: If you are installing e\*Index on a non-administrator workstation, you
should only select **e\*Index Global Identifier**.*

---

**5**   On the Select Components window, click **Next**.  The Select Program
Folder window appears.

*Next button*



Select a folder for
the program icons
on the *Select
Program Folder*
window
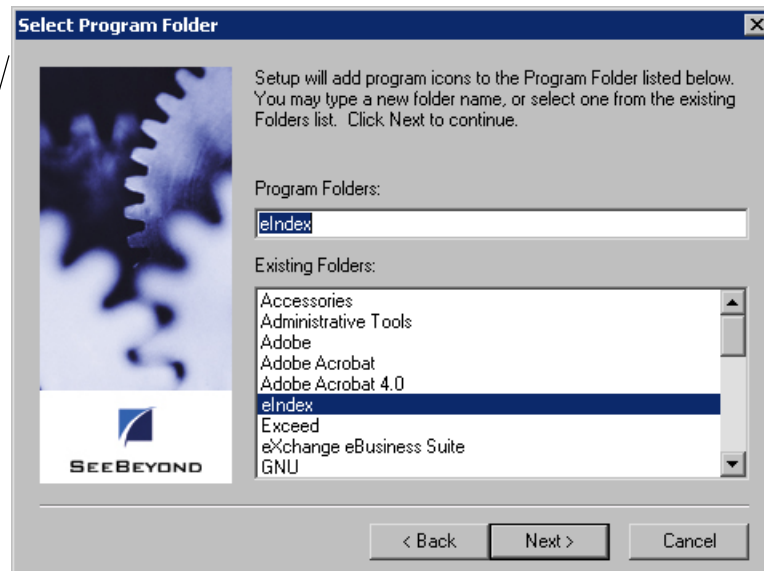
**6**   Specify the folder into which you want to install the program icons or
accept the default name, and then click **Next**.  The Check Setup
Information window appears.

*Next button*

Verify the installation information on the Check Setup Information window
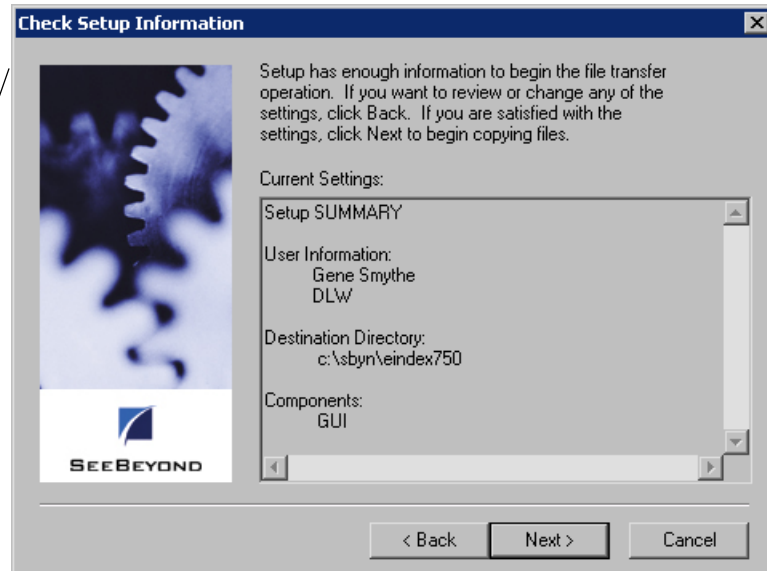
**7**  Verify the information you specified, and do one of the following:

*To change any of the options you selected*, click **Back**, and make the necessary changes.

*To continue with the installation*, click **Next**.  When the GUI files are installed, the Setup Complete window appears.
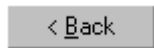
Back button

Next button

When the Setup Complete window appears, your installation is complete
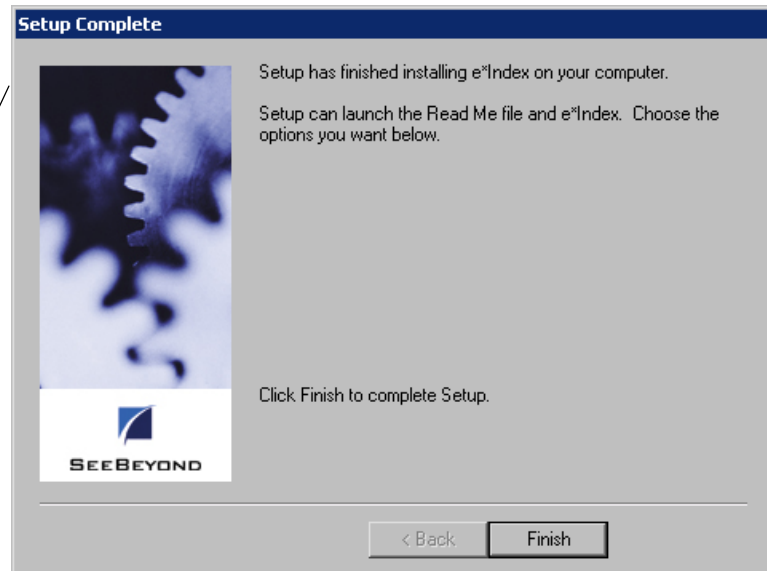
Finish button

**8**  Click **Finish** to conclude the installation process and return to the Windows desktop.

**9**  You can view the e*Index application icons on the Windows desktop.



e*Index
application
icons
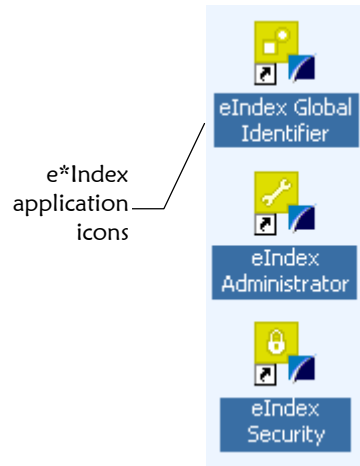
**10**  Continue to "Step 5: Copy the Publications."

## Step 5: Copy the Publications

The electronic documentation files for e*Index are located on the installation CD-ROM in the folder **\docs**.  Back up the files in your current e*Index publications directory, and copy the new files into the directory.  The following is a list of documents included in the electronic library:

■  **eI_411to452.pdf**
   The e*Index 4.1.1 to 4.5.2 Upgrade Guide describes how to upgrade all components of e*Index from version 4.1.1 to version 4.5.2.

■  **eI_installation_452.pdf**
   The e*Index Global Identifier Installation Guide explains how to install all components of e*Index, including the GUI, database, and e*Index schema.

■  **eI_reports_452.pdf**
   Working with Reports for e*Index Global Identifier describes the standard reports provided with e*Index, and includes instructions on running each report.

■  **eI_tech_ref_452.pdf**
   The e*Index Global Identifier Technical Reference is designed to assist e*Gate programmers in writing Monk scripts for the e*Ways for e*Index.

■  **eI_upgrade_452.pdf**
   The e*Index Global Identifier Upgrade Guide describes how to upgrade all components of e*Index from version 4.1.2 or higher to version 4.5.2.

■  **eI_userguide_452.pdf**
   The e*Index Global Identifier User's Guide explains how to use the e*Index GUI.

- **eIA_userguide_452.pdf**
  The e*Index Administrator User's Guide explains how to use the e*Index Administrator GUI.

- **eIS_userguide_452.pdf**
  The e*Index Security User's Guide explains how to set up and maintain security for the e*Index applications.

- **feedback_form.pdf**
  Use this form to provide any comments or suggestions for improving the documentation provided for e*Index.

- **init_load_452.pdf**
  This document describes how to perform a batch load of data from existing systems into the e*Index database.  This only needs to be performed when a new system is added to the e*Index network.

- **Readme.wri**
  This document provides information about the electronic library, such as using the cross-referencing index, search capabilities, and so on.

- **rel_notes_452.pdf**
  The e*Index Global Identifier Release Bulletin describes the changes made to the e*Index applications since e*Index 4.5.

- **rel_notes_411to452.pdf**
  The e*Index 4.1.1 to 4.5.2 Release Bulletin describes the changes made to the e*Index applications since e*Index 4.1.1.

- **Welcome_451.pdf**
  The Welcome Document lists all publications available in PDF format, and provides links to each document file.  Each file also links back to the Welcome Document.

- **UI_index.pdx**
  This is the index file that cross-references all PDF files in the electronic library, allowing you to search for information across documents.

- **UI_index (folder)**
  This folder provides the information required for the cross-referencing index to function.

## Step 6: Modify tnsnames.ora

The GUI client workstation must be able to connect to the e*Index database. To enable this, modify the **tnsnames.ora** file on the client machine by adding a new stanza the defines the e*Index 4.5.2 database.

*Note:  If you have both Oracle Server and Oracle Client installed, you may have two **tnsnames.ora** files.  In this case, you need to create identical database stanzas in both files for the new databases.*

## ▶ To modify tnsnames.ora

Before you begin:

✓ Complete "Step 5: Copy the Publications"

**1** On the machine from which you will run the migration package, navigate to the Oracle home directory and then to **\network\admin**.

**2** Make a backup of the file **tnsnames.ora**, and then open the file **tnsnames.ora** in any text editor.

**3** Create a new stanza for each database you just created, including the following information:

- The TNS service name of the database
- The connection protocol used to connect to the database
- The database host name or address
- The port number
- The Oracle SID name for your database

Following is a sample stanza from **tnsnames.ora**.  Your **tnsnames.ora** file may differ from this sample depending on how your Oracle networking is configured.

```
                        ei01.world =
                           (DESCRIPTION =
                             (ADDRESS_LIST =
                                 (ADDRESS =
                                   (PROTOCOL = TCP)
                                   (Host = 100.0.0.00)
                                   (Port = 1000)
                                 )
                             )
                             (CONNECT_DATA = (SID = EI01)
                             )
                           )
```

The TNS service name and domain of the database

The name or TCP/IP address of the database server

The Oracle SID name of the database
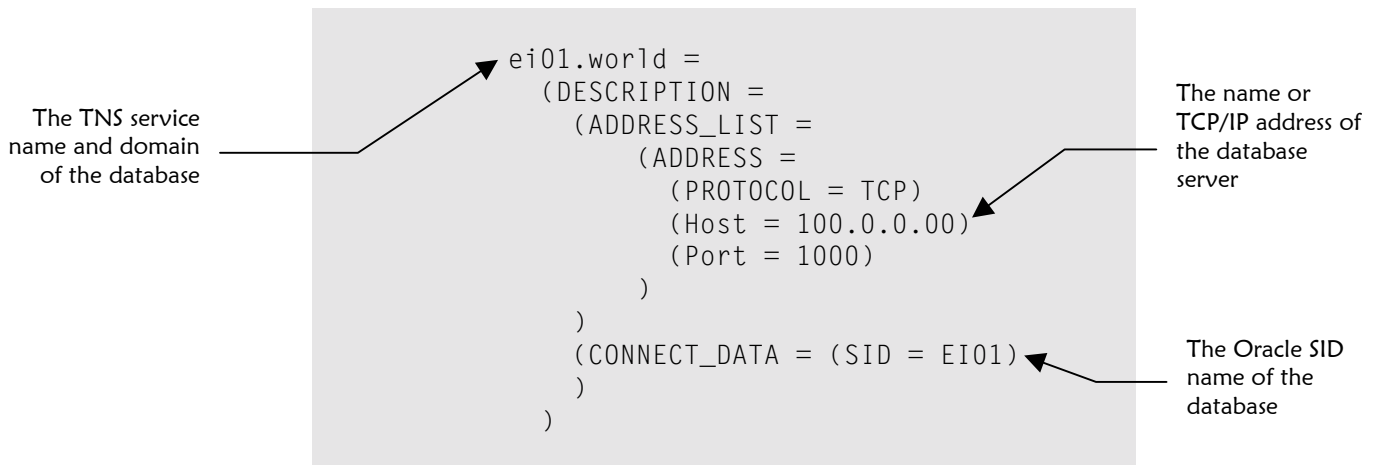
*Note:  For complete instructions on how to configure Oracle networking for your environment, refer to the appropriate Oracle user documentation.*

**4** Continue to "Step 7: Configure your 4.5.2 Target Database".

# Step 7: Customize stc_ua.ini

Before your e*Index GUI can connect to the database, you need to make sure that the e*Index initialization file contains a database definition for each database to which you need to connect through the e*Index GUIs.

## ▶ To modify stc_ua.ini

Before you begin:

✓ Step 6: Modify tnsnames.ora

**1** Navigate to the e*Index GUI home directory.

**2** In the **client** directory, make a backup copy of the file **stc_ua.ini**, and then open **stc_ua.ini** any text editor, such as Notepad or WordPad.

**3** Make the following modifications:

- In the **Database** stanza:

    – In the **DBMS** field, enter the appropriate database platform.  For Oracle, the DBMS is **084 ORACLE8i**.

    – In the **ServerName** field, enter the TNS service name for the database (found in the file **tnsnames.ora**).

    – In the **DisplayName** field, enter the name of the database as you want it to appear on the e*Index windows.

    ---

    *Notes:*

    ■ *If you are using more than one test database, you can add new database stanzas to **stc_ua.ini** with the additional database information.  The database numbers in the stanza headings must be sequential; for each stanza you add, make sure you increment the database number by one.*

    ■ *You do not need to enter any information in these fields: **LogID**, **LogPass**, **UserID**, **DBPass**, **Lock**, **DbParm**.  These fields are not currently used for e*Index initialization.*

    ---

- In the **STC** section:

    – In the **FacilityID** field, enter a valid system code.  This field cannot be left blank.  You can leave the default value (SBYN), however, for reporting purposes you may want to change the value to the system code of a system in your enterprise.

    – In the **DBIndex** field, enter the database number of the database you want as the default database on the login window.  The name of the database you specify automatically appears on the login window when you launch an e*Index application.  For example, to

specify the database with the stanza heading **Database4** as the default database, enter **4** in this field.

---

*Note:  The SmartCard section of **stc_ua.ini** only needs to be modified if you have implemented the SmartCard functionality of e\*Index.  Your SeeBeyond representative should perform the modifications.*

---

**4**   Continue to "Step 8: Register Online Help Support Files."

## Step 8: Register the Online Help Support File

If you haven't done so with previous versions, you should register the supporting help file before you use the online help system provided with the e\*Index GUIs.

### ▶ To register d2hPopup.ocx

Before you begin:

- ✓ Open MS-DOS from the Start menu

**1**   At the command prompt, navigate to your e\*Index home directory, and then to the **client** subdirectory.

**2**   At the prompt, type **regsvr32 d2hPopup.ocx**.

---

*Note:  If you do not register this file, you will be prompted to download the file from a website the first time you open an e\*Index online help file.  If you choose to download from the website, this file will be registered for you automatically.  Once this file is registered, you will not receive the prompt when you open the help files.*

---

## Step 9: Reboot the Computer

When you upgrade the e\*Index GUI, an environment variable, VTICFG, is created.  This variable tells the GUI the path in which the Vality rule set files should be downloaded from the database, and is set to "." (the current working directory).  In Windows 95 and 98, the installation adds this information as the last line in **autoexec.bat**, and creates a backup copy of the original **autoexec.bat** (named **autoexec.bak**) in case you need to revert to the previous version.  In order for this change to take effect on a Windows 95 or 98 computer, you need to reboot the machine.

# Migrating the Database

## About this Chapter

### Overview

This chapter presents the background information and the step-by-step instructions you need to migrate your e*Index database from version 4.1.1 to version 4.5.2.  This chapter also includes information about customizations you may need to make to your data both before and after the migration is complete.

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 Migration Files** | Learn about the files you need to modify and work with during the migration process |
| **Handling Errors** | Learn about the log files and database tables that report processing errors and how to fix those errors |
| **About the Installation** | Learn about the installation requirements and the files that are installed for the migration package |
| **Install the Migration Files** | Read step-by-step instructions for installing the database migration package |
| **Create the Databases** | Read step-by-step instructions for creating the databases you will use for the migration |
| **Migrate the Database** | Read step-by-step instructions for performing the database migration |

# What's Inside

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

# Learning About the Migration Files

## Overview

This section provides an overview of the properties and configuration files you need to work with to complete the database migration.  To upgrade the database from version 4.1.1 to 4.5.2, you need to migrate your existing 4.1.1 data into a new 4.5.2 database.  For more information about some of the issues you need to consider before migrating the database, see "Database Migration Considerations" in chapter 2, "Upgrade Process Overview".  For an overview of the database migration process, see "Database Migration Methodology" in chapter 2.

## About the Properties Files

The *properties files* define the runtime characteristics for the Java API used for the migration process.  There are several default properties files installed with the migration package.  You need to modify these files for your processing environment (this is described later in this chapter).  Make sure that the directory in which your customized property files are located is defined in the **CLASSPATH** environment variable on your computer.  The following properties files are used during the data migration.

- **Migration.properties**

- **411EiServer.properties**

- **45EiServer.properties**

- **CommonMigration.properties**

- **AuxMigration.properties**

### Migration.properties File

The **Migration.properties** file specifies information about the data migration and defines the names of the properties files used for each task.  All of the other properties files are referenced from the migration properties file, which is the only file that is used as a parameter for the migration Java commands.  The **Migration.properties** files contains these variables.

- **411EiServerPropFile**
  The name of the properties file that defines the attributes of the e*Index 4.1.1 source database (this database is the copy of your production database).  The default file is **411EiServer.properties.**

- **45EiServerPropFile**
  The name of the properties file that defines the attributes of the e*Index
  4.5.2 target database.  The default file is **45EiServer.properties**.

- **AuxMigrationPropFile**
  The name of the properties file that controls the auxiliary table migration.
  The default file is **AuxMigration.properties**.

- **DemographicAddressCode**
  The address type code to be assigned to each address in the database.
  e*Index 4.1.1 only stores one address for each person record, while
  e*Index 4.5.2 can store multiple addresses.  Each address in version 4.5.2
  is assigned an address type, such as Home, Work, Temporary, and so on.
  Typically, this value should be set to '**H**' for Home.

- **StartTime**
  A new start time for records to be processed in the case that a fatal error
  occurs in the middle of processing **Dm41ConvertHistory**.  For example, if
  the last record processed before a job halts has a create or update time of
  093002, you should specify that as the new start time when you restart the
  job.  The format for this variable is **HHmmSS**.

- **ProcessExceptions**
  An indicator that specifies whether the current run of
  **Dm41ConvertHistory** will process records from the *ui_history* table or if it
  will reprocess records from the exception tables that you have fixed.
  Enter **N** to process records from *ui_history*; enter **Y** to reprocess exception
  records.

- **EventUnmerge**
  The event type code for unmerge transactions.  In e*Index 4.1.1, "**U01**"
  was typically used for unmerge transactions.

- **EventMerge**
  The event type code for merge transactions.  In e*Index 4.1.1, "**A18**" was
  typically used for merge transactions.

- **CommonMigrationPropFile**
  The name of the file that controls the common table migration.   The
  default file is **CommonMigration.properties**.

- **DemographicWorkPhoneCode**
  The phone type code to be assigned to each work telephone number in
  the database.  e*Index 4.1.1 only stores one work telephone number for
  each person record, while e*Index 4.5.2 can store multiple numbers.  Each
  telephone number stored in version 4.5.2 is assigned a phone type, such
  as Home, Work, Fax, and so on.  Typically, this value should be set to
  "**CO**", which is the code for Business Telephone.

- **DemographicHomePhoneCode**
  The phone type code to be assigned to each home telephone number in

the database.  Typically, this value should be set to "**CH**" for Home Telephone.

■ **DateRange**
This is a performance parameter that allows you to specify the size of each block of data that is migrated. When data is retrieved from the 4.1.1 source database, it is retrieved in blocks in the size of the specified date range.  For example, if you specify 5, then history records within the first 5-day range are retrieved for the first block, records within the next 5-day range are retrieved for the second block, and so on.  Setting this to a small number is less efficient, but requires less heap space and database resources.  The value you should specify depends on the processing power of the migration server.

> *Note:  The **DateRange** variable does not control commits.  Records are committed to the 4.5.2 database at the end of each transaction.*

■ **StartDate**
The first create date or update date for the person and history records you want to process for a given run of the person data migration.  This variable, along with **EndDate**, allows you to perform the migration in stages, migrating only specific date ranges in one session.  If you perform the migration in stages, make sure you do it in chronological order, starting with the earliest create date in the database for your first history data migration run.  The format for the value of this variable is **YYYYMMDD**.

■ **EndDate**
The last create date or update date for the person and history records you want to process for a given run of the person data migration.  The format for the value of this variable is **YYYYMMDD**.

■ **UseClass1MiddleName**
An indicator that specifies whether to use the middle_initial field or the class1 field from *ui_person* in the 4.1.1 database to populate the middle_name field in the 4.5.2 database.  If you enter **N**, the middle_initial field in the 4.1.1 database is migrated to the middle_name field in the 4.5.2 database, and the class1 field is migrated to the class1 field.  If you enter **Y**, the class1 field is never populated in the 4.5.2 database, and the following occurs:

- If the 4.1.1 class1 field is populated, the 4.1.1 class1 data is migrated to the 4.5.2 middle_name field and any data in the 4.1.1 middle_initial field is lost.

- If the 4.1.1 class1 field is not populated and the 4.1.1 middle_initial field is populated, the 4.1.1 middle_initial is migrated to the 4.5.2 middle_name field.

## *EiServer.properties Files

There are two **\*EiServer.properties** files: **411EiServer.properties** and **45EiServer.properties**.  These files define properties about the 4.1.1 source database, which is the copy you will make of your production database, and the 4.5.2 target database, respectively.  They define information about the databases, such as the server and database name, the login ID and password, the database port, and so on.  They also define processing parameters, such as maximum query sizes, cache sizes, tracing levels, and so on.  The **\*EiServer.properties** files define the following variables.  Two variables,  the locations of the Vality name matching and address parsing rules, are only defined in the **45EiServer.properties** file.

- **databaseTimeOut**
  The maximum length of time in seconds that a database statement can be processed before an exception is thrown.  Configure this variable to ensure that processes that take a long time do not degrade system performance.

- **databaseMaximumQuerySize**
  The maximum number of records that can be retrieved during a weighted person search. Configure this variable to ensure that searches do not return large result sets, which could slow down performance.

- **databaseMaximumConnections**
  The maximum number of database connections that may be used at one time.

- **databaseMaximumStatementCacheSize**
  The maximum number of prepared statements that can be cached per connection.

- **databaseNetworkProtocol**
  The network protocol for the connections to the database.  For Oracle, this can be set to any protocol supported by Net8.  It is only required for the JDBC OCI driver.

- **databaseDriverType**
  The Oracle JDBC driver type.  The possible values are **thin** and **oci8**.

---

*Important!  For the following six variables, make sure that you enter information about the 4.1.1 source database in the **411EiServer.properties** file, and about the 4.5.2 target database in the **45EiServer.properties** file.*

---

- **databaseUserId**
  The user name with which to log on to the e\*Index database.  The user ID specified must be defined in e\*Index security, and must have the appropriate access permissions assigned.

- **databasePassword**
  The password associated with the specified **databaseUserId**.

- **databasePortNumber**
  The port number where a server is listening for requests from the Java program.

- **databaseServerName**
  The name of the database server on which the e*Index database is running.  This value can be the name or IP address of the server.

- **databaseVendor**
  The database platform being used for the e*Index database.  Currently only Oracle is supported for the migration package.

- **databaseName**
  The Oracle SID name of the database to which you want to connect.

- **matchNameServiceId**
  The Vality rule set to use for creating phonetic codes.  The default value, **1**, specifies that the default rule set, **ENCODE**, will be used.  Rule sets are defined in the file **MatchCfgs.cfg**, located in the **\config** subdirectory.  For more information about this file, see "About the Configuration File" in this chapter.  This variable is only in the **451EiServer.properties** file.

- **matchAddressServiceId**
  The Vality rule set to use for parsing addresses.  The default, **2**, specifies that the United States address rule set will be used.  To specify that the Australia address rule set will be used, change this value to **3**.  This variable is only in the **451EiServer.properties** file.

- **traceLevel**
  The level of tracing to be performed.  The possible values for this variable are TRACE, INFO, WARNING, or ERROR.  If you specify TRACE, then the log files will grow to be very large during the migration process, but the logged information will be very thorough.

## CommonMigration.properties

This file defines which database tables are migrated when you migrate the common code tables.  Modifying this file allows you to control which of the following tables are migrated:  *ui_ethnic*, *ui_event*, *ui_language*, *ui_suffix*, *ui_mstatus*, *ui_dept*, *ui_status*, *ui_sex*, *ui_race*, *ui_system*, *ui_veteran*, *ui_vip*, *ui_religion*, *ui_title*, *ui_state*, and *ui_country*.  You can perform multiple runs of the auxiliary table migration, migrating different tables with each run.  To specify that a table be migrated in a specific run, set the value of that table to **true**.  To specify that a table not be migrated in a specific run, set the value of that table to **false**.

SeeBeyond recommends that you do not migrate any table that you have not customized, since the default data has been improved in some of these tables.  If you do not migrate tables whose data has been customized, you must define the custom elements manually using e*Index Administrator 4.5.2.  If

you are unsure whether a table has been customized, you should migrate the table.

### AuxMigration.properties

This file defines which database tables are migrated when you migrate the auxiliary tables. Modifying this file allows you to control which of the following tables are migrated: *ui_audit*, *ui_comment*, *ui_message*, *ui_facility*, *ui_canned_msg*, *ui_control, ui_ctrl_rule, ui_ctrl_file, control_sec, ui_nickname*, and *ui_zip*. Note that *ui_ctrl_rule* and *ui_ctrl_file* must be migrated together. You can perform multiple runs of the auxiliary table migration, migrating different tables with each run. To specify that a table be migrated in a specific run, set the value of that table to **true**. To specify that a table not be migrated in a specific run, set the value of that table to **false**.

SeeBeyond recommends that the *ui_message* table not be migrated due to changes in terminology between versions 4.1.1 and 4.5.2. SeeBeyond also recommends that you do not migrate any table that you have not customized, since the default data has been improved in some of these tables. Exceptions to this are the *ui_audit*, *ui_comment*, and *ui_ctrl_rule_file*. If you do not migrate *ui_facility*, *ui_control*, or *control_sec*, you must define systems and control key values manually using the e*Index 4.5.2 GUIs.

## About the Configuration File

The configuration file defines the rule sets provided with the migration package and allows you to specify the address rule set files to use. The configuration file provided with the migration package is named **MatchCfgs.cfg**, and is located in the **\config** subdirectory. This file contains a stanza for each Vality rule set, including the rule set for phonetic encoding (**ENCODE**) and the rule sets for parsing addresses (**USADDR** and **AUADDR**). Note that no matching is performed during the migration since all records existing in the production database have already been identified and matched. The standard name-matching rule set, **UI**, is not used during the migration and is replaced by the **ENCODE** rule set, which simply creates phonetic codes.

The **matchNameServiceId** and **matchAddressServiceId** variables in the migration properties file specify which of the rule sets in the configuration file to use. The rule sets are specified by number, and the stanzas are numbered in order sequentially with the first stanza being **1**. Following is a sample stanza for a rule set file in **MatchCfgs.cfg**.

```
CONFIG .
KEY ENCODE.DCT
STAN ENCODE.STN
RECLEN 350
```

The variables included in each stanza are as follows:

- **CONFIG**
  This variable specifies the name of the rule set. By default, the name of the rule set used for phonetic coding in e*Index is named **ENCODE**. Two additional rule sets for address parsing are also provided. The USADDR rule set is for United States addresses; the AUADDR rule set is for Australia addresses.

- **KEY**
  This variable specifies the name of the match key dictionary file. This file always has an extension of **.DCT**.

- **STAN**
  This variable specifies the name of the standardization file. This file always has an extension of **.STN**.

- **RULES**
  This variable specifies the name of the rules file. This file always has an extension of **.RUL**.

- **RECLEN**
  This variable specifies the record length for messages processed by the Vality matching algorithm.

## Log Files

Each time you execute a Java command for the data migration a log file is created in the directory from which you executed the command. The log file is named **dm<date_time>.log**, where <date_time> is the date and time the log file was created. Log files record useful information about the records processed during the migration, and can help pinpoint errors or warnings. You can control the amount of information logged by changing the **traceLevel** variable in the **\*EiServer.properties** files. The trace level specified in **411EiServer.properties** controls logging for transactions against the 4.1.1 source database; the trace level specified in **45EiServer.properties** controls logging for transactions against the 4.5.2 target database. The trace levels are:

- **Trace**
  This level logs extensive details of the operation (a significant amounts of information). The log files can grow quite large when this level is used.

- **Info**
  This level logs basic status information, such as startup or parameter changes.

- **Warning**
  This level logs problems that do not cause operation failure and faults that were satisfactorily handled.

■ **Error**
This level logs problems that may cause a given operation to fail, but are not serious enough to cause the module itself to halt.

# About Processing or Data Errors

## Overview

During the migration process, you may encounter some data in the e*Index database that contains anomalies and cannot be processed into the 4.5.2 target database.  In this case, you can view the records that could not be processed, read information about the errors that occurred, and then repair the data and reprocess the records.

## About the Exception Tables

Once you create your e*Index 4.1.1 source database, you can run a SQL script against the database to create exception tables that maintain information about the migration and provide information about specific records that cannot be migrated due to errors.  The following tables are created by the SQL script.  A diagram of the tables appears in Figure 5-1 on page 5-13.

- ◼ **dm_except_header**
  This table provides information about each error that occurred while migrating person data, including the UID of the record, the event type, the update date and time, and whether the  error was resolved.  This table also includes a column that indicates whether each record should be reprocessed.

- ◼ **dm_except_detail**
  This table provides a complete description of the error that occurred for each record in the *dm_except_header* table.  This table references IDs from the *dm_except_header* and *dm_except_def* tables.

- ◼ **dm_module**
  This table lists the different modules in which an error can occur, such as person add, alias add, person merge, security, and so on.  This table is referenced by the *dm_except_def* table described below.

- ◼ **dm_except_def**
  This table lists the different types of errors that can occur, and references the *dm_module* table for the module in which each error occurs.  This table is referenced by the *dm_except_detail* table, which lists the ID of the error type for each record.

- ◼ **dm_aux_except**
  This table provides information about errors that occur when migrating data in the code tables, including a complete description of each error.  It references the *dm_except_def* table for the error type.

**Figure 5-1: Migration Exception Tables Diagram**
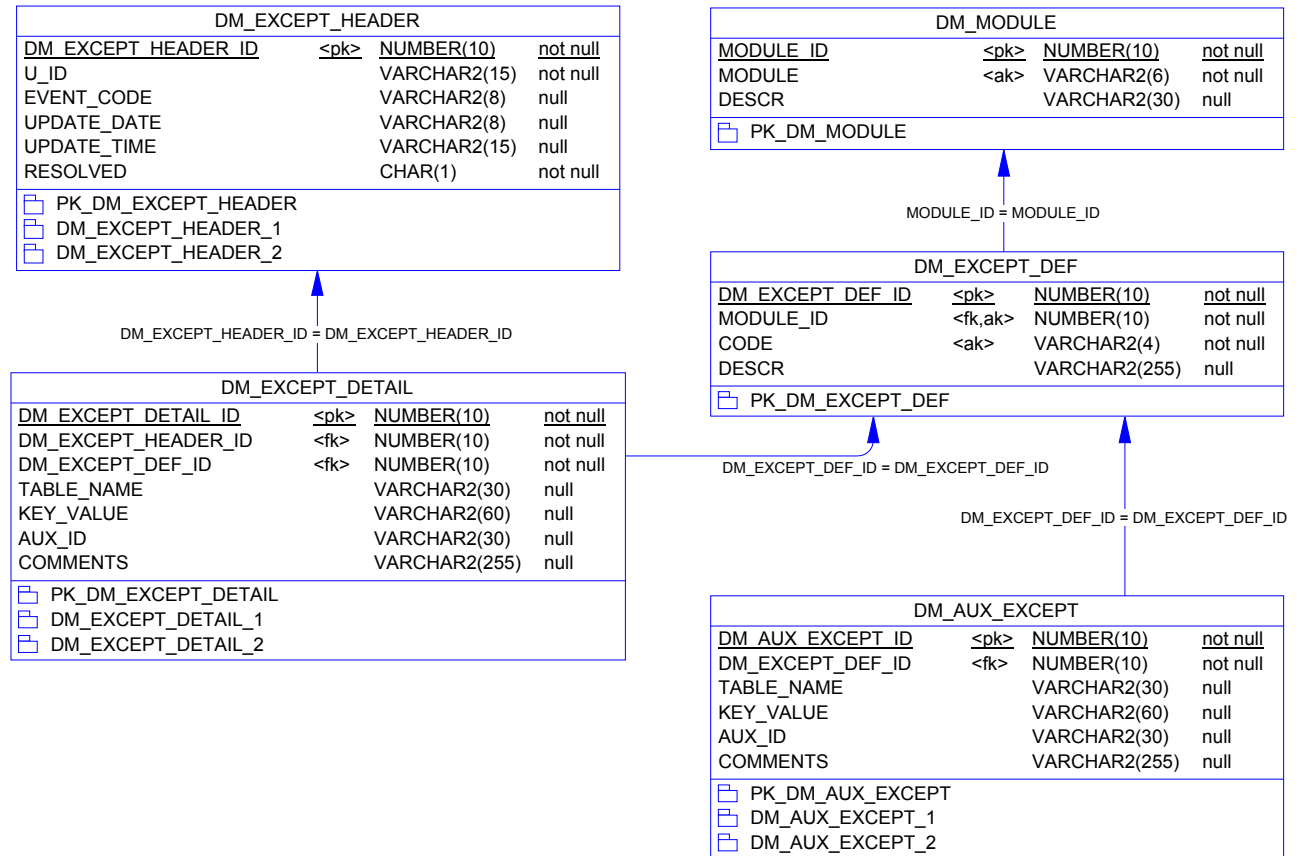


## Correcting Exception Records

When records cannot be processed and are written to the exception tables, you need to fix the errors in order to reprocess those records. The exception tables should provide enough information for you to determine where the error occurred. When fixing the exception records, be sure not to modify the following fields:

- In the *ui_history* table:
  - update_date
  - update_time
  - update_function

- In the *ui_local_id* table:
  - create_date

- create_time

■ In the *ui_alias* table:

- create_date

- create_time

- update_function

Note that the u_id field should never be changed in any table.

Some examples of errors that may occur include records associated with facility codes that are not defined in the e*Index database, invalid null values, unique key constrain violations, invalid local ID or person status, and so on. The descriptions provided by the exception tables should give you a good idea of where the data error occurred.

# Learning About the Migration Package Installation

## Overview

This section of the chapter provides background information about installing the database migration package and the environment variables you need to modify.

## Getting Started

Installing and working with the migration package for the database requires changes to the computing environment on the machine from which the migration will be performed.  SeeBeyond recommends installing the migration package on the database server for the 4.5.2 target databases.  The following hardware and software must be installed in order to perform the migration.

- **Database server:**
  SeeBeyond recommends that the migration be performed on a separate database server from your production database, and that the new database server become your production server once the migration is complete.  This server should have processing, memory, and networking configurations that are similar to or better than the production server.

- **Supported operating systems:**
  Windows NT 4.0 with service pack 6  (SP6), Windows 2000, HP Unix 10.20 or later, IBM AIX 4.3, Compaq TRU64 4.0 or later, or Solaris 2.6 or later

- **Required database software:**
  Oracle Server version 8.1.7.  We recommend that the migration process be performed from the same server where 4.5.2 target database resides.  If you choose to run the migration package on a client workstation instead of the database server, you need to install Oracle Client version 8.1.7 on the client workstation.  A standard installation of Oracle should include the JDBC™ libraries, which are required for the migration (these files are located in the Oracle home directory in **\jdbc\**).

- **Required Java software:**
  Java™ 2 SDK, Standard Edition 1.1.8 or later (for HP Unix 10.20, you must install version 1.1.8; for HP Unix 11.0, you must install version 1.2 or later; for all other platforms, version 1.3.1 is recommended)

## Platform Information

The installation procedures for the migration package on Windows NT and UNIX differ slightly, but both installations begin on a machine running Windows 95, Windows 98, or Windows NT 4.0 SP 4 or later, or Windows 2000. For a Unix installation, you can either install the files on a Windows server and copy them to the Unix server, or you can map a network drive to the Unix server and specify that drive in the installation path.

When you install the migration package for the database, the files are created in several subdirectories in the path you specify for the installation. One of the subdirectories, named "**bin**", contains libraries that are specific to each operating system. These files are separated into subdirectories labeled by the operating system for which they should be used. You can simply copy the files for your operating system into the **\bin** directory, and delete the rest. This is described in "Step 4: Move the Migration Package Files" later in this chapter.

## Directory and File Structure

When you install the migration package, library, database, configuration, and Java files are installed in the installation path. The installation also provides batch files for you to export and import your 4.1.1 production database. After you install the migration package, the directory and file structure should resemble the illustration in Figure 5-2 on the following page. In this image, the variable **<install_dir>** represents the path you specify in the InstallShield. If you install the package on the server on which you will be using them, **<install_dir>** is also the migration package home directory. After installation, you need to move some of the files in the **\bin** directory. Figure 5-3 on page 5-18 illustrates the structure of the home directory after you have moved the required files as described later in this chapter in "Step 4: Move the Migration Package Files". In this diagram, **<home_dir>** is the home directory of the migration package.

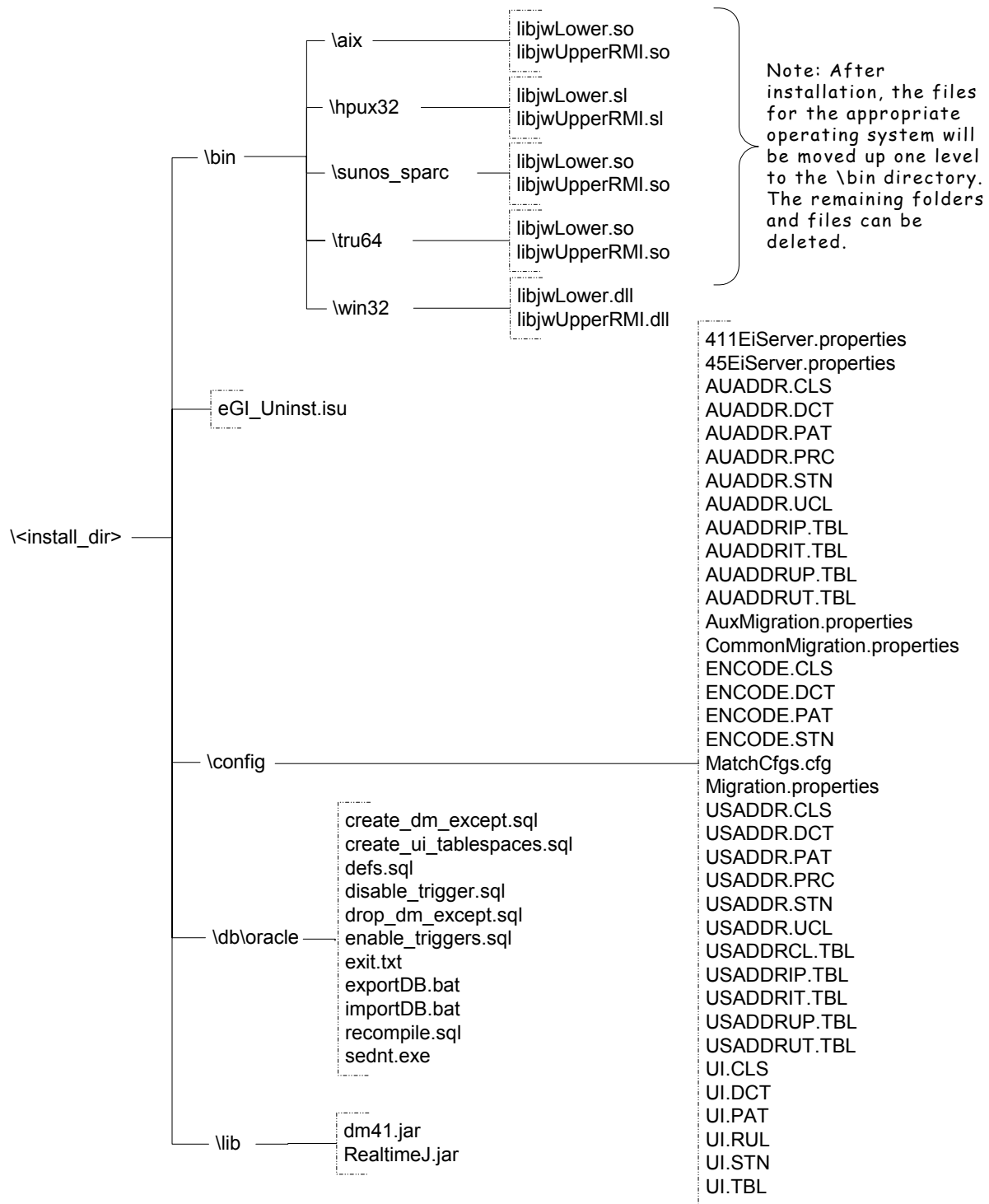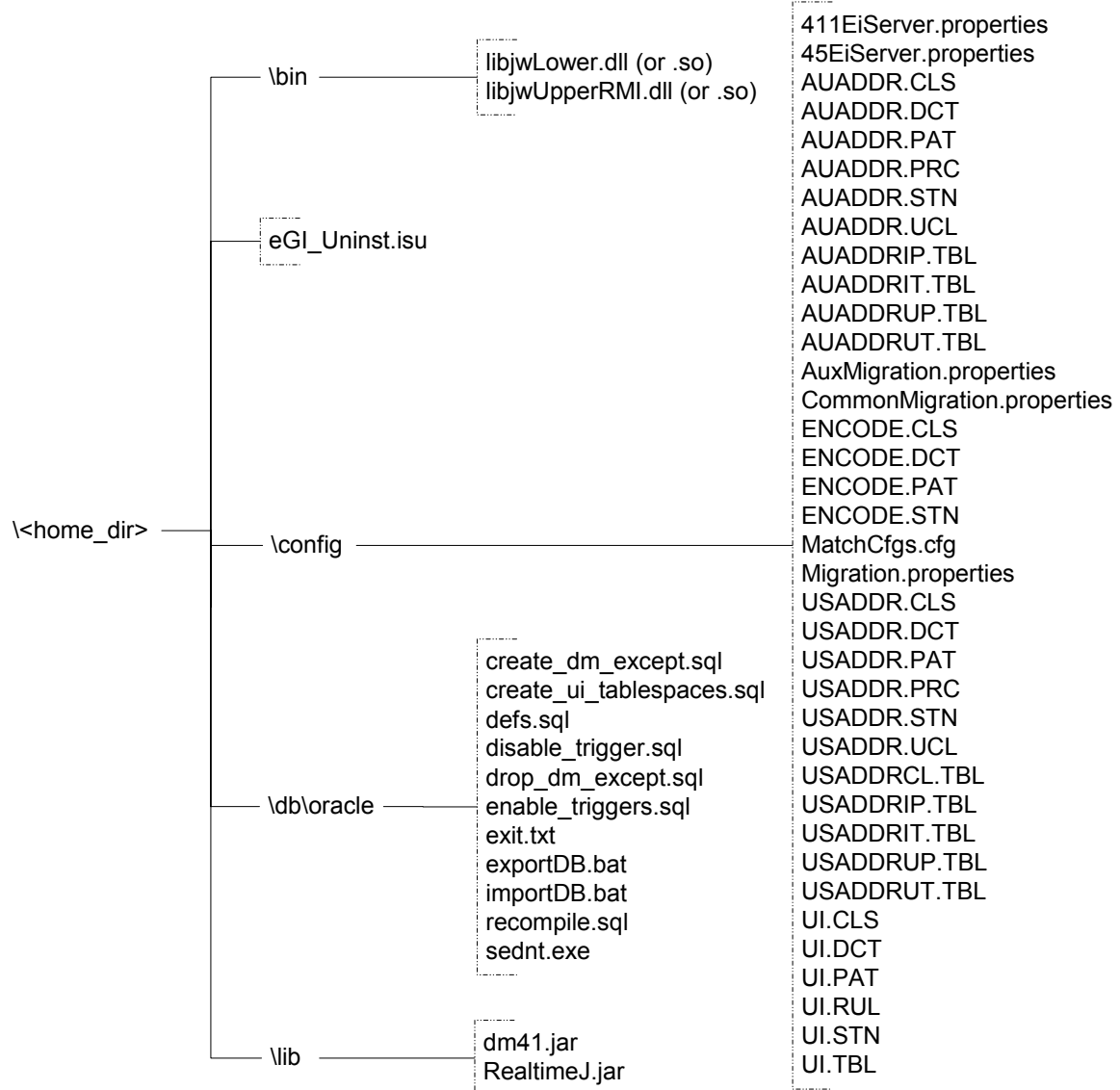**Figure 5-2: Migration Package Directory and File Structure**

```
\<install_dir> ─┬─ \bin ─────┬─ \aix ──────────┬─ libjwLower.so
                │            │                 └─ libjwUpperRMI.so
                │            │
                │            ├─ \hpux32 ───────┬─ libjwLower.sl
                │            │                 └─ libjwUpperRMI.sl
                │            │
                │            ├─ \sunos_sparc ──┬─ libjwLower.so
                │            │                 └─ libjwUpperRMI.so
                │            │
                │            ├─ \tru64 ────────┬─ libjwLower.so
                │            │                 └─ libjwUpperRMI.so
                │            │
                │            └─ \win32 ────────┬─ libjwLower.dll
                │                              └─ libjwUpperRMI.dll
                │
                ├─ eGI_Uninst.isu
                │
                ├─ \config ───── 411EiServer.properties
                │                45EiServer.properties
                │                AUADDR.CLS
                │                AUADDR.DCT
                │                AUADDR.PAT
                │                AUADDR.PRC
                │                AUADDR.STN
                │                AUADDR.UCL
                │                AUADDRIP.TBL
                │                AUADDRIT.TBL
                │                AUADDRUP.TBL
                │                AUADDRUT.TBL
                │                AuxMigration.properties
                │                CommonMigration.properties
                │                ENCODE.CLS
                │                ENCODE.DCT
                │                ENCODE.PAT
                │                ENCODE.STN
                │                MatchCfgs.cfg
                │                Migration.properties
                │                USADDR.CLS
                │                USADDR.DCT
                │                USADDR.PAT
                │                USADDR.PRC
                │                USADDR.STN
                │                USADDR.UCL
                │                USADDRCL.TBL
                │                USADDRIP.TBL
                │                USADDRIT.TBL
                │                USADDRUP.TBL
                │                USADDRUT.TBL
                │                UI.CLS
                │                UI.DCT
                │                UI.PAT
                │                UI.RUL
                │                UI.STN
                │                UI.TBL
                │
                ├─ \db\oracle ── create_dm_except.sql
                │                create_ui_tablespaces.sql
                │                defs.sql
                │                disable_trigger.sql
                │                drop_dm_except.sql
                │                enable_triggers.sql
                │                exit.txt
                │                exportDB.bat
                │                importDB.bat
                │                recompile.sql
                │                sednt.exe
                │
                └─ \lib ─────── dm41.jar
                                RealtimeJ.jar
```

Note: After installation, the files for the appropriate operating system will be moved up one level to the \bin directory. The remaining folders and files can be deleted.

**Figure 5-3: Migration Home Directory and File Structure**

The diagram below illustrates the structure of the migration package once all files have been copied or moved as required.

```
                                    \bin ──────── libjwLower.dll (or .so)      411EiServer.properties
                                                  libjwUpperRMI.dll (or .so)   45EiServer.properties
                                                                               AUADDR.CLS
                                                                               AUADDR.DCT
                                                                               AUADDR.PAT
                                    eGI_Uninst.isu                             AUADDR.PRC
                                                                               AUADDR.STN
                                                                               AUADDR.UCL
                                                                               AUADDRIP.TBL
                                                                               AUADDRIT.TBL
                                                                               AUADDRUP.TBL
                                                                               AUADDRUT.TBL
                                                                               AuxMigration.properties
                                                                               CommonMigration.properties
                                                                               ENCODE.CLS
                                                                               ENCODE.DCT
                                                                               ENCODE.PAT
                                                                               ENCODE.STN
\<home_dir> ──── \config ─────────────────────────────────────────────────── MatchCfgs.cfg
                                                                               Migration.properties
                                                                               USADDR.CLS
                                                                               USADDR.DCT
                                    create_dm_except.sql                       USADDR.PAT
                                    create_ui_tablespaces.sql                  USADDR.PRC
                                    defs.sql                                   USADDR.STN
                                    disable_trigger.sql                        USADDR.UCL
                                    drop_dm_except.sql                         USADDRCL.TBL
                    \db\oracle ──── enable_triggers.sql                        USADDRIP.TBL
                                    exit.txt                                   USADDRIT.TBL
                                    exportDB.bat                               USADDRUP.TBL
                                    importDB.bat                               USADDRUT.TBL
                                    recompile.sql                              UI.CLS
                                    sednt.exe                                  UI.DCT
                                                                               UI.PAT
                                                                               UI.RUL
                                    dm41.jar                                   UI.STN
                    \lib ───────── RealtimeJ.jar                               UI.TBL
```

# Installing the Migration Package

## Overview

To install the database migration package, you must complete the following steps:

- Step 1: Install the Java Environment
- Step 2: Install or Upgrade Oracle
- Step 3: Install the Migration Package
- Step 4: Move the Migration Package Files
- Step 5: Modify Environment Variables

*Note: SeeBeyond recommends that the migration package be installed on, or copied to, the same server on which you will install the 4.5.2 target database.*

## Step 1: Install the Java Environment

Before you can work with the Java commands in the migration package, you need to install Java™ 2 SDK, Standard Edition on the machine on which you will perform the migration. Java™ 2 SDK is a development environment for building applications and components that can be deployed on the Java platform. Much of the migration is performed using Java commands. Make sure you install the Java Developer's Kit (JDK), Java Runtime Environment (JRE), and Java Virtual Machine (JVM). Follow the instructions provided with the Java™ 2 SDK to install the application. For more information about working with Java™ 2 SDK, refer to the appropriate Java user documentation.

## Step 2: Install or Upgrade Oracle

If you run the migration package from the machine on which the new e*Index 4.5.2 database will reside, you need to install Oracle 8.1.7 Server on that machine. If you run the migration package from a client workstation, the package requires Oracle 8.1.7 Client in order to connect with the e*Index databases. Either way, you must install one of the Oracle Java Database Connectivity (JDBC) drivers to connect to the e*Index database. You can use either the Oracle JDBC thin driver or Oracle JDBC/OCI driver. Make sure you install the appropriate version for the version of Java JDK you are using. If you install the OCI driver, remember to configure the **tnsnames.ora** file for the e*Index database. For complete instructions on installing and configuring Oracle, see the appropriate Oracle8*i* documentation. SeeBeyond recommends that you install Oracle 8.1.7.2.1.

## Step 3: Install the Migration Package

To begin the installation process, insert the e*Index 4.5.2 CD-ROM into the CD-ROM drive on your computer, and make sure no other Windows applications are running.

### ▶ To install the migration package

Before you begin:

✓ Complete "Step 1: Install the Java Environment" and "Step 2: Install or Upgrade Oracle"

✓ Close all Windows applications

**1** Insert the e*Index 4.5.2 CD-ROM into the CD-ROM drive of your computer.  For Unix installations, you can either install the files to a Windows computer and copy the files to the Unix server, or map a network drive to the Unix server and specify the mapped drive as your installation path during the installation.

**2** On the CD-ROM drive, navigate to the **\migration** directory and then double-click **Setup.exe**.  The Welcome window appears, reminding you to close all Windows programs.

The migration package is installed using a standard InstallShield Wizard
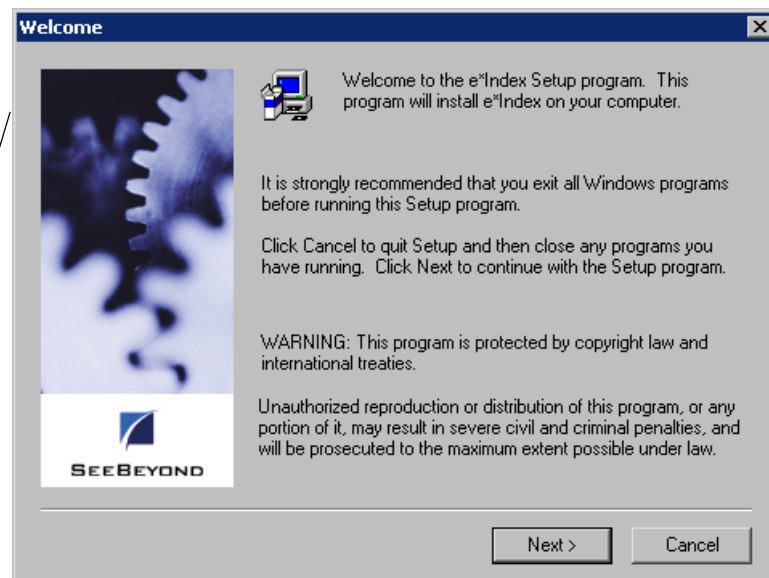
**3** On the Welcome window, do one of the following:

*To close any open Windows programs*, click **Cancel**, close the programs, and then repeat step 1.

Cancel button

*To continue with the installation process without closing any external programs*, click **Next**.  The Software License Agreement window appears.

Next button

You must agree
to the terms of
the license
agreement in
order to proceed

**4** If you agree to the license agreement, click **Yes**. The Information window
appears with a list of requirements to remind you of the applications you
need to install before installing e*Index.

*Yes button*
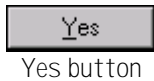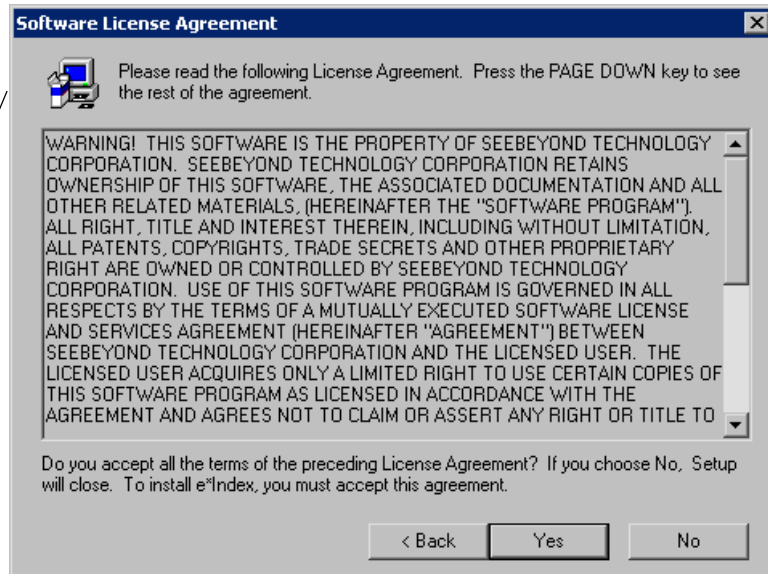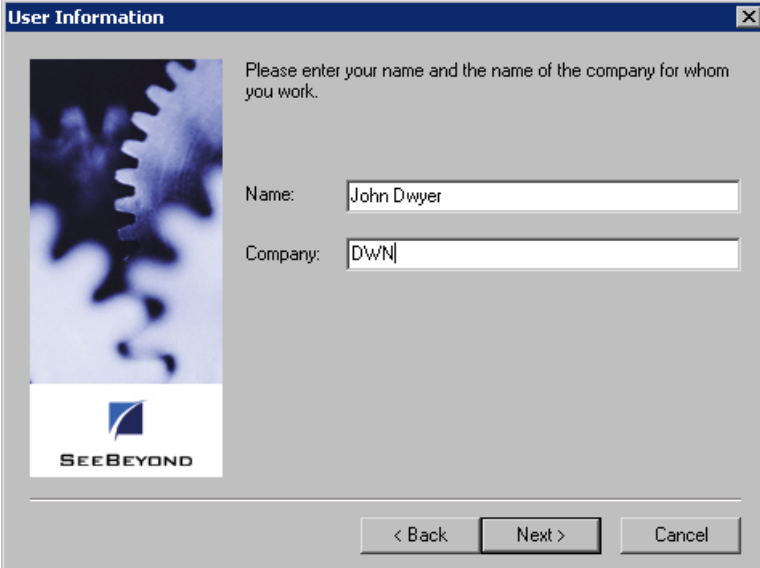
The Information
window reminds
you of the system
requirements
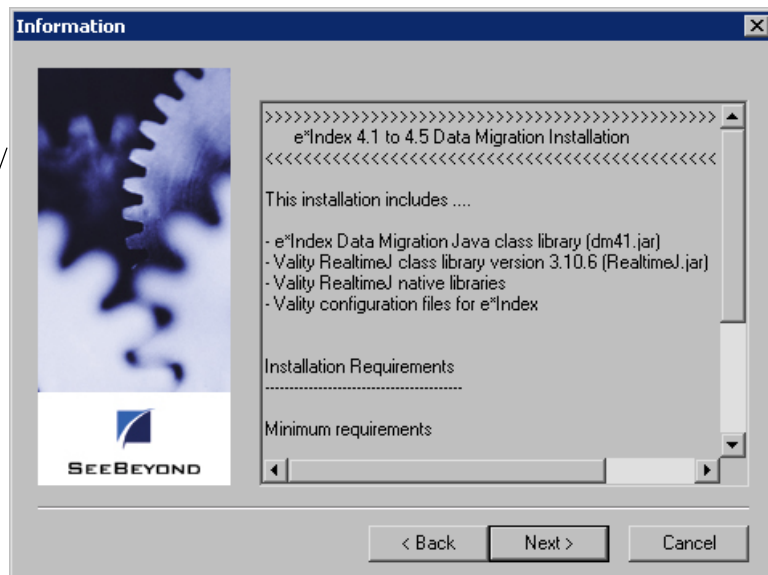
**5** On the Information window, click **Next**. The User Information window
appears.

*Next button*

Enter your name
and your company
name on the User
Information
window
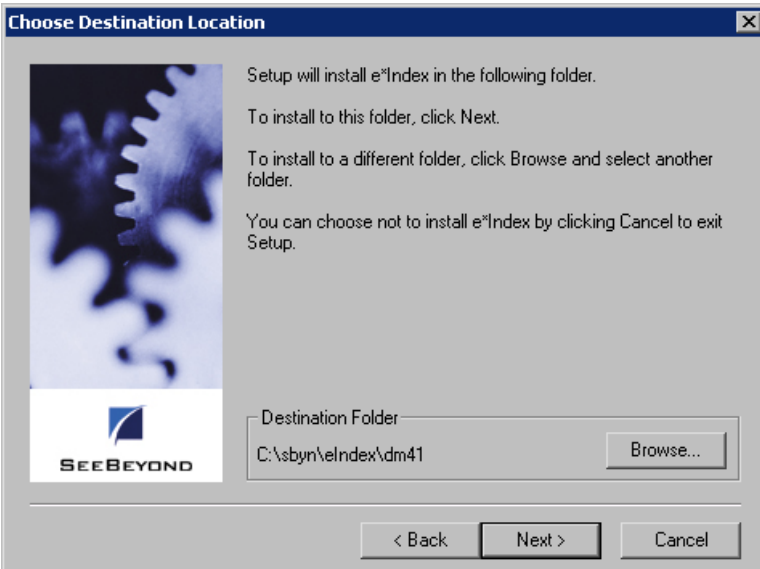
Next >

Next button

**6** On the User Information window, enter your name and your company's name, and then click **Next**.  The Choose Destination Location window appears.



On the Choose
Destination
Location window,
verify the
installation path for
the migration files

**7** On the Choose Destination Location window, do one of the following:

Next >

Next button

*To install the files in the folder that appears in the Destination Folder path*, click **Next**.

Browse...

Browse button

*To change the location in which the files will be installed*:

- Click **Browse**.
- On the Choose Folder dialog, type or select the path where you want to install the files.

Specify a new
installation path on
the Choose Folder
dialog
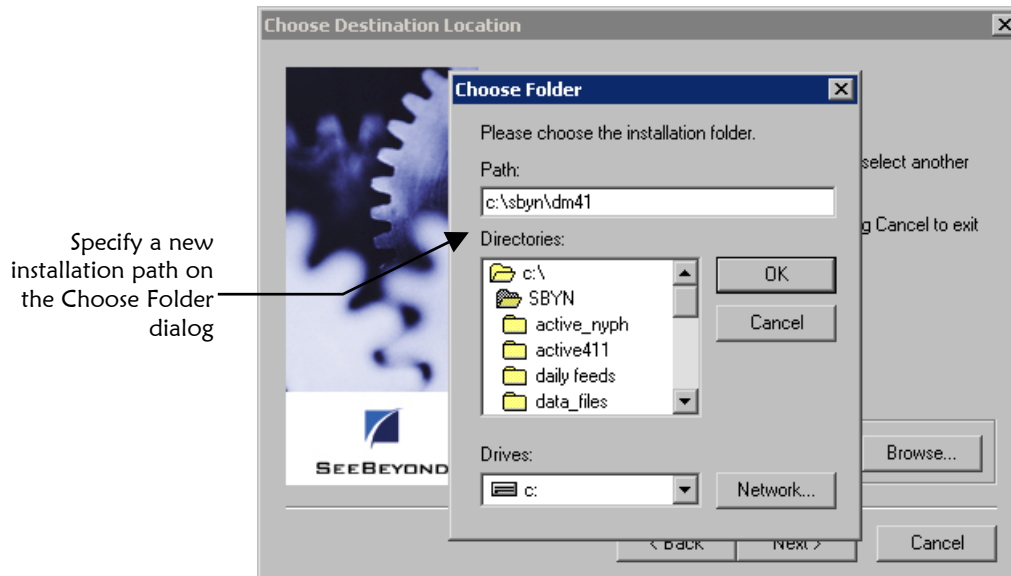
- Click **OK**. The new path you specified appears in the Destination Folder path.

*Note:  If the path you specified does not exist, a dialog appears asking if you want to have the folder created.  If you select **Yes**, Setup creates the specified path for you.*

Next >

*Next button*

**8**  On the Choose Destination Location window, click **Next**. The Check Setup Information window appears.



The Check Setup
Information
window displays
the options you
have chosen
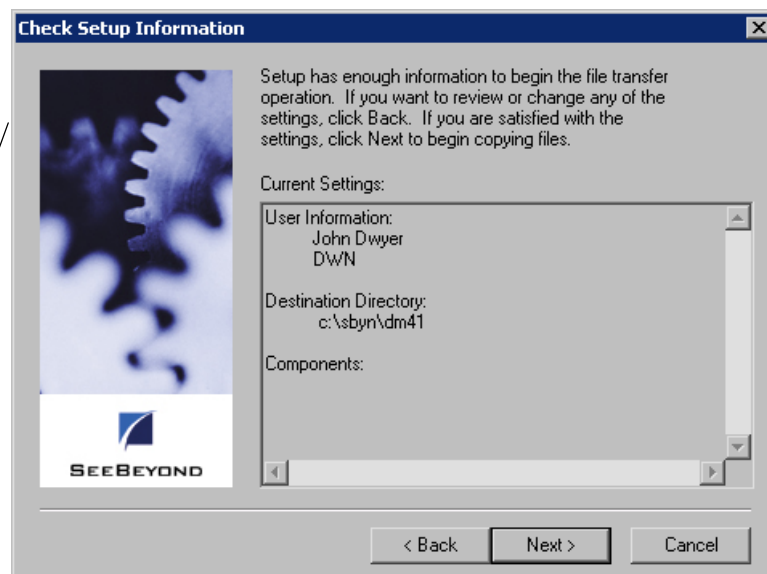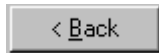
< Back

Back button

Next >

Next button

**9** On the Check Setup Information window, verify the information you specified, and do one of the following:

*To change any of the options you selected*, click **Back**, and make the necessary changes.

*To install the files in the specified directory*, click **Next**.  The Setup Complete window appears after the files are installed.

The Setup Complete window indicates that the files have been installed

**Setup Complete**

Setup has finished installing e*Index on your computer.

Setup can launch the Read Me file and e*Index.  Choose the options you want below.

Click Finish to complete Setup.

**SeeBeyond**
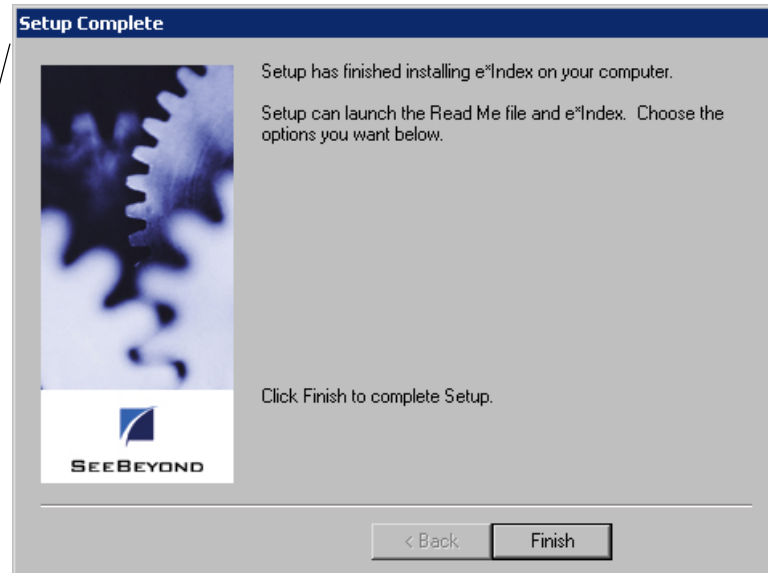
< Back    Finish

Finish

Finish button

**10** The Setup Complete window indicates that all necessary files are installed.  On the Setup Complete window, click **Finish** to conclude the setup process.

**11** Continue to "Step 4: Move the Migration Package Files".

## Step 4: Move the Migration Package Files

If you did not install the migration files on the machine on which you will run the migration, you need to move the files to that machine.  Before you move the files, create the home directory for the files on the migration host machine.  Once you create the home directory on the machine that will host the migration package, you can move the files to that machine.  If you installed the files on the migration server, you only need to move the files from the **\bin\<os>** directory as described in step 3 below.

▶ **To move the migration package files**

Before you begin:

✓ Complete "Step 3: Install the Migration Package"

> *Note:*  *In the steps below, **<install_dir>** refers to the path in which you installed the files, **<home_dir>** refers to the new home directory you created for the files, and **<os>** refers to the operating system of the host machine.  If you installed the package on the migration server, **<install_dir>** and **<home_dir>** are the same path.*

**1**   If you are moving the files to a different server, create a home directory for the files on the new server.  For example:

```
\eIndex\migration
```

**2**   Move all of the folders located in the **<install_dir>** on the installation server to the **<home_dir>** on the migration server.

**3**   Move the files located in **\<home_dir>\bin\<os>** to **\<home_dir>\bin**, where **<os>** is the operating system running on the migration package host server.  If you installed the files on the host server, you only need to move these files up one directory so they reside in **\bin**.  After you move the files, you can delete the **<os>** folders.

**4**   Verify the directory structure of the home directory.  The new directory structure should resemble Figure 5-3 on page 5-18.

**5**   Continue to "Step 5: Modify Environment Variables".

# Step 5: Modify Environment Variables

In order to work with the Java commands for the migration package, certain environment variables need to be set on the migration server.  Make sure to set all of the variables described in the following instructions before working with the Java migration commands.

## ▶ To modify environment variables

Before you begin:

✓   Complete "Step 4: Move the Migration Package Files"

> *Note:*  *In addition to the variables described below, make sure to set the environment variables for your Java SDK environment as described in your Java documentation.  If you have more than one Java SDK environment installed, make sure the JAVA_HOME variable is set to the environment you want to use.*

**1**   On the machine where the home directory resides, add the **\<home_dir>\bin** path to your library path environment variable.  The variable you need to modify is specific to the operating system you are using.

   •   For Windows, modify the **PATH** variable
   •   For Sparc Solaris, modify the **LD_LIBRARY_PATH** variable

- For HP Unix, modify the **SHLIB_PATH** variable
- For AIX, modify the **LIBPATH** variable

**2**   Add a new environment variable named **VTICFG** and define the value as the absolute path of the **\<home_dir>\config** directory.

> *Important!  If you are running the migration on a machine that already hosts an e\*Index GUI environment, the VTICFG variable will already be defined.  If you change the path, remember that the rule set files in the specified path will be used by the GUI and the Java API. When you launch the GUI, rule set information from the database is downloaded into the path specified by VTICFG, overwriting the existing files.*

**3**   Add the following information to the **CLASSPATH** variable:

- The absolute path and filename of the **classes12.zip** file in your Oracle environment.  This file should be located in the **\jdbc\lib** subdirectory of your Oracle home directory.

- The absolute path and filenames of the migration package **.jar** files. The **.jar** files are named **dm41.jar** and **RealtimeJ.jar** and are located in the **\<home_dir>\lib** directory.

- The path to the properties files used by the migration package.  These files are located in the **\<home_dir>\config** directory.

**4**   If you use the Oracle OCI/JDBC driver, you also need to add the following paths to the PATH variable:  **\<Oracle_home>** and **\<Oracle_home>\bin** (where **<Oracle_home>** is the Oracle home directory).

**5**   Make sure your Java SDK variables are set up as required for the version of Java you are using (see your Java documentation for more information).

> *Note:  Oracle JDBC uses the JNDI package, which was an extension in Java 1.2, but became part of the core in version 1.3.  If your Java environment is version 1.2.x or earlier, you need to add the JNDI package to your CLASSPATH.  This is not required for version 1.3.*

# Creating the Migration Databases

## Overview

The database migration is not performed against your production database. Instead, you use a copy of the production database to migrate the data into a new e*Index 4.5.2 database and then switch the new database over to production.  Complete the following steps to create the migration databases.

- ■  Step 1:  Obtain Database Information

- ■  Step 2:  Install or Upgrade Oracle Server

- ■  Step 3:  Create the Source Database

- ■  Step 4:  Create the e*Index 4.5.2 Target Database

- ■  Step 5:  Modify **listener.ora**

- ■  Step 6:  Modify **tnsnames.ora**

- ■  Step 7: Configure the 4.5.2 Target Database

- ■  Step 8: Remove Extraneous Code Table Data

- ■  Step 9: Begin Queuing 4.1.1 and 4.5.2 Daily Transactions

- ■  Step 10:  Merge or Resolve Potential Duplicate Records

- ■  Step 11:  Back up the Production Database

- ■  Step 12: Export the Production Database

- ■  Step 13: Restart the e*Index 4.1.1 e*Ways

- ■  Step 14: Import the Production Data into the Source Database

## Step 1: Obtain Database Information

Before beginning, obtain information about the databases you will be working with.  The migration uses three databases:  the production database (to export data only), a copy of the production database (the "source" database), and a new 4.5.2 target database.  For each database, you need to know the database server name, the database home directory, and the Oracle TNS and SID names.  Make sure the SID names you select for the new databases are different from any existing SID names in your Oracle installation.  To reduce network latency, SeeBeyond recommends running the migration program from the machine that hosts the 4.5.2 target database.

Before beginning the database installations, analyze the database size requirements for the databases.  The tablespace sizes and extents must be

specified before running the installation scripts.  Your copy of the production databases should have the same size tablespace files, the same block sizes, and so on.  Your 4.5.2 database should be of equivalent size, though the files and indexes are different.

Finally, before beginning this process, analyze your configuration requirements for the 4.5.2 database, GUIs, and processing.  Determine the values for the new control keys, any display customizations you need to make, country-specific options to be modified, query configurations, and so on.  You should also determine whether you will migrate the old security information or recreate the security configuration.  For a complete list of new customizations, see the *e\*Index 4.1.1 to 4.5.2 Release Bulletin*.  "Database Migration Considerations" also includes information about functions that are new to version 4.5.2.

## Step 2: Install or Upgrade Oracle Server

Before you install the e\*Index database files, Oracle 8.1.7 Server must be installed on the database server where the new databases will reside.  When you install Oracle, make a note of the path in which you install the program.  You will need to specify this information later in the database installation files.  For information about installing or upgrading Oracle Server, refer to the appropriate installation documentation for Oracle.

If you are installing the databases in a Unix environment, you will need to execute one file (**create2.bat**) for the 4.5.2 target database from a client workstation.  You must install Oracle 8.1.7 Client on the machine from which you will run **create2.bat**.  If you run the data migration from a client workstation instead of the database server, you must have Oracle 8.1.7 Client installed on that machine as well.  For performance purposes, SeeBeyond recommends that you run the migration files from the database server.

---

***Important!***  *It is crucial that the Oracle Server installation is operational prior to performing the following steps.*

---

## Step 3: Create the Source Database

The migration package provides a batch file that will modify an existing Oracle database to make an exact copy of the production database, but you need to create the database structure first.  You can create a standard Oracle database using either standard Oracle tools or by running the file **create1**(**.bat**) from the e\*Index 4.1.1 database installation scripts.  The instructions for running **create1** are located in chapter 3 of the *e\*Index Global Identifier Installation Guide* for e\*Index 4.1.1.  If you still have the installation scripts you used for your production database, you can modify those scripts to create your new database by changing the database name and pathnames.  Make sure to match the sizing attributes of your production database, and

use the same character set for the source database as was used for the old.  If you are not sure which character set your production database uses, type this command at the SQL prompt to display the character set type:

```
select * from v$nls_parameters
where parameter='NLS_CHARACTERSET';
```

# Step 4: Create the e*Index 4.5.2 Target Database

The e*Index 4.5.2 target database will become your production database once all data is migrated.  You can install this database on the same server that hosts the 4.1.1 source database if the server's processing capabilities are powerful enough.  Install this database using the database installation scripts provided in the e*Index 4.5.2 InstallShield.  For instructions on installing a 4.5.2 database, refer to chapter 3 of the *e*Index Global Identifier Installation Guide* for version 4.5.2.  Make sure to specify sizing attributes for the new database that are equivalent to the sizing attributes of your production database.

# Step 5: Modify listener.ora

Before the migration files can connect to the new databases, you need to customize **listener.ora** and then restart it.  This file is located in the Oracle home directory in the **\network\admin** subdirectory on your migration database server.

▶ **To modify listener.ora**

Before you begin:

✓ Complete "Step 3: Create the Source Database" and "Step 4: Create the e*Index 4.5.2 Target Database"

**1** On the database server, navigate to the Oracle home directory, open the **\network\admin** subdirectory, and then make a backup copy of **listener.ora**.

**2** Open the file **listener.ora** in any text editor.

**3** Append a **SID_DESC** stanza to the **SID_LIST_LISTENER** section at the end of the file, entering your SID name as illustrated below.  Your file may differ slightly from this example depending on how your Oracle environment is configured.  Do this for both of the databases you just created.

```
            SID_LIST_LISTENER =
              (SID_LIST =
                (SID_DESC =  …
                )
                (SID_DESC =
                  (GLOBAL_DBNAME = ei01.world)
                  (ORACLE_HOME = C:\Orant)
                  (SID_NAME = EI01)
                )
              )
```

Add the database
SID name to
listener.ora

*Note:  If the first section of the file, the Listener description list, does not contain information for the database server(s), you should enter that information now.  Refer to the appropriate Oracle documentation to configure this section.*

**4**   Save and close **listener.ora**.

**5**   Stop and then restart Listener.  To do this in a Windows environment, open the Control Panel, select **Services**, and then stop and restart the Listener service.  In Unix, use the utility **lsnrctl** to stop and restart Listener.

**6**   Continue to "Step 6: Modify tnsnames.ora".

## Step 6: Modify tnsnames.ora

The machine from which you will run the migration package must be able to connect to the new databases you created.  To enable this, modify the **tnsnames.ora** file on the migration machine so there are two new stanzas, one for the 4.1.1 source database and one for the 4.5.2 target database.  If you are unsure of how to modify the Oracle **tnsnames.ora** file, refer to the appropriate Oracle8*i* documentation, or refer to "Step 6: Modify **tnsnames.ora**" in chapter 4 of this guide.

*Note:  If you have both Oracle Server and Oracle Client installed, you may have two **tnsnames.ora** files.  In this case, you need to create identical database stanzas in both files for the new databases.*

## Step 7: Configure the 4.5.2 Target Database

Before you begin the actual data migration, you should configure certain attributes of the 4.5.2 environment.  The only 4.5.2 attributes specifically used in the migration are the address parsing rules; however, performing all necessary customizations at this time will reduce the amount of down-time

when moving the 4.5.2 database to production.  Most of these customizations are optional except where noted.

▶ **To configure the 4.5.2 target database**

  ✓ Complete "Step 6: Modify tnsnames.ora"

  ✓ Make sure the e\*Index Administrator 4.5.2 GUI is installed (for more information, see chapter 4, "Upgrading the GUIs")

**1** Log on to the e\*Index Administrator GUI.

**2** Modify control key values for the new control keys.  These include ALSRCHLMT, COUNTRY, MIXEDCASE, PVSUMMARY, SHORTID, and UVAUDITLOG.  For the migration, the COUNTRY control key must be set to either **USA** or **AUS**.

---

*Important!  The country control key must be set to the value that corresponds to the address parsing rule set you are using for the migration.  For the **USADDR** rule set, specify **USA**.  For the **AUADDR** rule set, specify **AUS**.*

---

**3** Configure country-specific options.  For more information, see "Configuring Country-Specific Options" in chapter 5 of the *e\*Index Administrator User's Guide*.

---

*Important!  Any changes to the address parsing options must be made before proceeding with the data migration.*

---

**4** Specify the fields that appear on the e\*Index GUIs , the fields that are hidden, the fields that are required, and the labels for each field.  For more information, see "Configuring Display Options" in chapter 5 of the *e\*Index Administrator User's Guide*.

**5** Customize the configurable query.  For more information, see "Configuring Queries" in chapter 5 of the *e\*Index Administrator User's Guide*.

---

*Note:  If you currently have the EXTNSVSRCH control key set to **Y**, you must modify the configurable query for default and phonetic searches in order for those searches to be set for extensive searching.  For more information about extensive searching, see "Extensive Searching" in chapter 2 of this guide and in the* e\*Index 4.1.1 to 4.5.2 Release Bulletin.  *For information about how to configure the query for extensive searching, see "Configuring Extensive Searching" in chapter 5 of the* e\*Index Administrator User's Guide.

---

**6** If you installed region-specific security in the 4.5.2 database, you must define regions in the Common Table Maintenance function.  For more information, see chapter 3 of the *e\*Index Administrator User's Guide*.

7   Define any common table data elements that are new in e*Index 4.5.2. These data types include Address Type, Citizenship, Districts of Residence, Driver License Issuer, Event Notification, Nationality, Person Category, and Phone Type. For more information, see chapter 3 of the *e*Index Administrator User's Guide*.

8   Define non-unique ID types. For more information, see "Non-Unique ID Types" and "Step 3: Create Non-Unique ID Types" in chapter 3 of the *e*Index Administrator User's Guide*.

9   If you are not migrating the security tables, set up security at this time by creating user profiles and user groups, assigning access permissions, assigning users to regions (if you are using region-specific security), and assigning user profiles to user groups. For more information, see the *e*Index Security User's Guide*. If you are migrating security information, do not modify 4.5.2 security information at this time.

10  Continue to "Step 8: Remove Extraneous Code Table Data".

## Step 8:  Remove Extraneous Code Table Data

Before you migrate any of the common table or auxiliary table data, you may need to remove existing rows from certain database tables in the 4.5.2 database. If you changed any of the processing codes for any of the code tables you are migrating (for example, you changed the codes used for languages or religions), you should remove the existing records in *stc_common_detail* for that data type. For more information about the common tables you need to migrate, see "Database Migration Considerations" in appendix A of this guide.

### ▶ To remove extraneous code table data from the 4.5.2 database

Before you begin:

✓ Complete "Step 7: Configure the 4.5.2 Target Database"

✓ Review the Administrative Table Data Mapping Chart in appendix A of this guide

1   Referring to the Administrative Table Data Mapping Chart, determine the type of code table data you need to delete from the 4.5.2 target database.

2   Open a SQL editor, such as SQLPlus or SQLPlus Worksheet, connecting to the 4.5.2 target database.

**3**   For each common code table you need to migrate, type the following command:

**delete from stc_common_detail where common_header_id=<id_no>**

where <id_no> is the common header ID for the type of data being deleted as defined in the *stc_common_header* table.  These IDs are listed for each data type in the Administrative Table Data Mapping Chart in appendix A.

**4**   For each auxiliary code table you need to migrate, type the following command:

**truncate table <table_name>**

where <table_name> is the name of the table containing the data you want to delete.  The Administrative Table Data Mapping Chart provides a list of 4.5.2 table names for auxiliary code tables.

**5**   When you have removed all the data of the types you are migrating, commit the changes to the database.

**6**   Continue to "Step 9: Begin Queuing 4.1.1 and 4.5.2 Daily Transactions".

# Step 9: Begin Queuing 4.1.1 and 4.5.2 Daily Transactions

To ensure that no transactions are lost during the migration process, begin queuing the transactions being sent into the e*Index Schema from external systems before completing the following steps.  By this time, the e*Index 4.5.2 Schema should be tested and ready for production, and you need to queue records from external systems into both the 4.1.1 and 4.5.2 Schemas.  This will allow you to continue to use the 4.1.1 database while at the same time queuing up the live data for the 4.5.2 database.

# Step 10: Merge or Resolve Potential Duplicate Records

During the migration process, the data in the *ui_duplic* table is not converted to the new database.  To maintain the integrity of the e*Index data, review your current potential duplicate list and resolve or merge any records flagged as potential duplicates before you export the production data.  If you do not clean out the potential duplicate table, all of the potential duplicate information will be lost in the migration process.  Make sure that no e*Ways are processing data into the database and no client workstation users are connected to the database before you begin the potential duplicate cleansing.

## Step 11: Back up the Production Database

Prior to exporting the migration data from your e*Index production database, you should make a complete backup of your current database. For more information on performing Oracle database backups, see the appropriate Oracle documentation.

## Step 12: Export the Production Database

Database export and import batch files are provided with the e*Index database migration package in order to simplify the process for you and to ensure that all the required database objects are created in the 4.1.1 source database.

### ▶ To export the production database

Before you begin:

✓ Complete "Step 11: Back up the Production Database"

**1** From a Unix or MS-DOS command line, navigate to the home directory of the migration package, and then to **\db\oracle**.

**2** At the command prompt, type the following command:

**exportDB.bat system <up> <tnsname> <dmp_file>**

where:

- <up> is the password for the "system" user logon ID
- <tnsname> is the Oracle TNS service name of the 4.1.1 production database
- <dmp_file> is the name of the export file (the file name should have an extension of **.dmp** per Oracle naming conventions)

**3** This process creates a log file of the export process named **export.log** in the **\db\oracle** directory.  Review this file to verify the export process had no errors.  The database export file is also created in this directory.

*Note:  You may see warnings in the export log regarding information that cannot be read.  These warnings do not cause any data issues and can be ignored.*

**4** Continue to "Step 13: Restart the e*Index e*Ways".

## Step 13: Restart the 4.1.1 e*Index e*Ways

The migration of data from the source to the target database may take several days to several weeks, depending on the size of your production database.

During this time, you can use your current 4.1.1 production database while at the same time queuing transactions in the 4.5.2 Schema. This allows you to maintain a usable e*Index database throughout the migration. Begin processing data back into the 4.1.1 production database by restarting the e*Index e*Ways as soon as you complete the export process. The new transactions being processed through the 4.1.1 production database at this time will not need to be migrated to the 4.5.2 database since all of the same transactions are also being stored in the 4.5.2 queue.

---

*Important!* *From this point on, all transactions are being fed into the 4.1.1 database and stored in the 4.5.2 queue. In order to ensure that both sets of data match after the migration, do not make any changes to the 4.1.1 database through the GUI workstation, SQLPlus, or the initial load program.*

---

## Step 14: Import the Production Data into the Source Database

In order to create an exact replica of your production database, you need to import the data exported from the production database into the source database you created in "Step 3: Create the Source Database". An import batch file was created for this purpose.

▶ **To import the production data into the source database**

Before you begin:

✓ Complete "Step 13: Restart the e*Index e*Ways"

**1** Navigate to the home directory of the migration package, and then to **\db\oracle**. Open the file **defs.sql** in any text editor.

**2** Modify **defs.sql** for the source database you created in "Step 3: Create the Source Database".

---

*Tip:* *If you are unsure of how to modify* **defs.sql***, see "Step 6: Modify defs.sql" in chapter 3 of the* e*Index Global Identifier Installation Guide *for e*Index 4.1.1. You can also refer to the* **defs.sql** *file you used to create your production 4.1.1 database.*

---

**3** Create a user named "UI" in the source database. To do this, connect to Oracle as **SYS** or **Internal** using SQLPlus, and then type the following command.

> **create user ui identified by ui;**
> **grant dba to ui with admin option;**
> **commit;**

**4** From a Unix or MS-DOS command line, navigate to the home directory of the migration package, and then to **\db\oracle**.

**5** At the command prompt, type the following command to run the initial phase of the import:

> **importDB.bat system <up> <tnsname> <dmp_file>**

where:

- <up> is the password for the "system" logon ID

- <tnsname> is the Oracle TNS service name of the 4.1.1 source database

- <dmp_file> is the name of the export file created from the **exportDB.bat** process

**6** When the initial import is complete, review the log files created in the **\db\oracle** directory for any errors that may have occurred.  The log files are named **import.log**, created by the Oracle import utility, and **master_import.log**, which logs information about the import process.

---

*Notes:*

- ■ *The **master_import.log** file will contain errors about objects that already exist.  These errors are expected and will not affect the integrity of the data.*

- ■ *You may see several warnings in about unrecognized statements in the export file and about objects created with compilation warnings.  These warnings do not cause any data issues and can be ignored.  They should not appear if you are using Oracle 8.1.7.2.1.*

- ■ *To verify that all potential duplicate records were resolved or merged, check that the* ui_duplic *table is empty in the source database.*

---

# Migrating the Database

## Overview

To migrate the data from your source 4.1.1 database to the target e*Index 4.5.2 database, you need to complete the following steps.  Make sure to carefully follow the instructions provided.

- Step 1: Modify the Database Properties Files

- Step 2: Modify the Migration Properties File

- Step 3: Copy Address-parsing Rule Set Files

- Step 4: Create and Populate Exception Tables

- Step 5: Disable 4.5.2 Database Triggers

- Step 6: Migrate Security and Common Tables

- Step 7: Migrate Auxiliary Tables

- Step 8: Migrate Person Data

- Step 9: Review and Reprocess Exception Records

- Step 10: Migrate Assumed Match Data

- Step 11: Review the Migration Process

- Step 12: Enable 4.5.2 Database Triggers

- Step 13: Load the Production Vality Rule Set

- Step 14: Assign a Region to each System (optional)

- Step 15: Update Security Configuration

- Step 16: Catch up from e*Gate Queue

## Step 1: Modify the Database Properties Files

Before you can use the Java commands to migrate the database information, you need to specify certain information about each database in the properties files for each database.  There are two database properties file: one for each database you created for the migration.

### ▶ To modify the database properties files

Before you begin:

- ✓ Make sure you have completed all of the procedures in the preceding section "Creating the Migration Databases"

**1**  Navigate to the migration package home directory, and then to **\config**.

**2**  Open the file **45EiServer.properties** in any text editor and modify the variables in the file for the 4.5.2 target database.  For a complete list of variables and their definitions, see "*EiServer.properties Files" on page 5-4.

**3**  Save and close **45EiServer.properties.**

**4**  Open the file **411EiServer.properties** in any text editor and modify the variables in the file for the 4.1.1 source database.

**5**  Save and close **411EiServer.properties**.

**6**  Continue to "Step 2: Modify the Migration Properties File".

## Step 2: Modify the Migration Properties File

The migration properties file specifies information for the migration such as the names of the database, auxiliary, and common properties files, event codes, whether to use the middle initial or middle name, and so on.

### ▶ To modify the migration properties files

Before you begin:

✓  Complete "Step 1: Modify the Database Properties Files"

**1**  Navigate to the migration package home directory, and then to **\config**.

**2**  Open the file **Migration.properties** in any text editor and modify the variables in the file.  For a complete list and description of the variables, see "Migration.properties File" on page 5-4.

**3**  Save and close **Migration.properties**.

**4**  Continue to "Step 3: Copy Address-parsing Rule Set Files".

## Step 3: Copy Address-parsing Rule Set Files

Unlike e*Index 4.1.1, e*Index 4.5.2 uses two Vality rule sets; one for name matching and one for address parsing.  While you can use the default Vality rule set files for testing the migration program, you need to make sure the Vality files located in the migration home directory contain the production address rule set files for version 4.5.2 when you run the actual migration (the name-matching rule set is not used during the migration since matching has already been perform on the existing data).

▶ **To copy address-parsing rule set files**

Before you begin:

✓ Complete "Step 2: Modify the Migration Properties File"

✓ Obtain the production version of the Vality rule set files for address parsing

**1** Copy the production version of the address parsing rule set files to **\<home_dir>\config**.   Copy all of the **USADDR.\*** files for United States installations, or all of the **AUADDR.\*** files for Australia installations.

---

*Important!  If you are using the Australian address rule set, make sure to change the value of **matchAddressServiceId** in the **45EiServer.properties** file to **3**.  For US installations, leave the value at its default, **2**.*

---

**2** Continue to "Step 4: Create and Populate Exception Tables".

## Step 4: Create and Populate Exception Tables

The migration package includes scripts that create exception tables in the 4.1.1 source database.  These tables store information about any errors that occur while the migration is processing, including the UIDs of the records that could not be processed and the reason for the error.   Another SQL script, **drop_dm_except.sql**, is provided in case you need to drop the exception tables.

▶ **To create and populate exception tables**

Before you begin:

✓ Complete "Step 3: Copy Address-parsing Rule Set Files"

✓ Make sure the e*Index 4.1.1 source database is started

**1** From a Unix or MS-DOS command line, navigate to the home directory of the migration package, and then to **\db\oracle**.

**2** At the command prompt, type the following command:

   **sqlplus <un>/<up>@<db_name> @create_dm_except**

where

• <un> is your login ID for the e*Index 4.1.1. source database
• <up> is the password for the given login ID
• <db_name> is the Oracle TNS name of the e*Index 4.1.1 source database

**3** At the command prompt, type the following Java command:

> **java com.stc.eIndex.migration.Dm41InsertExceptionDefs Migration.properties**

**4**   When the Java process is done running, continue to "Step 5: Disable the Target 4.5.2 Database Triggers".

# Step 5: Disable the Target 4.5.2 Database Triggers

For performance reasons, you need to disable certain triggers in the target database.  A SQL script is provided for this purpose.

▶ **To disable the target 4.5.2 database triggers**

Before you begin:

- ✓  Complete "Step 4: Create and Populate Exception Tables"
- ✓  Make sure the e*Index 4.5.2 target database is started

**1**   From a Unix or MS-DOS command line, navigate to the home directory of the migration package, and then to **\db\oracle**.

**2**   At the command prompt, type the following command:

> **sqlplus <un>/<up>@<db_name> @disable_triggers**

where

- •  <un> is your login ID for the e*Index 4.5.2. target database
- •  <up> is the password for the given login ID
- •  <db_name> is the Oracle TNS name of the e*Index 4.5.2 target database

**3**   Continue to "Step 6:  Migrate Security and Common Tables".

# Step 6:  Migrate Security and Common Tables

The migration is performed in steps, with the common and security tables being migrated first.  Common tables include certain data types that are populated through the e*Index Administrator (previously e*Index Data Dictionary) in the Common Maintenance function, such as religions, marital statuses, languages, countries, and so on.  You can specify whether to migrate individual common tables, but if you migrate security information all security tables are migrated.  There is no required order for migrating these tables.

### ▶ To migrate security and common tables

Before you begin:

- ✓ Complete "Step 5: Disable the Target 4.5.2 Database Triggers"
- ✓ Refer to the Code Table Migration Chart in chapter 2 for information about the common tables you need to migrate

**1** Navigate to the migration package home directory, and then to **\config**.

**2** Open the file **CommonMigration.properties** in any text editor.

**3** This file lists all of the 4.1.1 common tables that can be migrated. To specify that a table be migrated, set the value to **true**. To specify that a table *not* be migrated, set the value to **false**.

*Important!*

■ *You can perform this portion of the migration in steps, migrating only a few tables at a time, or you can migrate all tables at once. You should not migrate any tables in which you have not added any custom information or modified existing records.*

■ *Between 4.1.1 and 4.5.2, terminology changed from "system" to "source" and from "facility" to "system". The* ui_system *variable in this file refers to 4.1.1 systems and not to facilities. Facilities are migrated in the following step.*

■ *If you choose to migrate the* ui_state *table, the create dates for the records will default to 1901-01-01 since no date exists in the 4.1.1 database.*

**4** Save and close **CommonMigration.properties**.

**5** Run the following Java command:

> **java com.stc.eIndex.migration.Dm41ConvertAdmin Migration.properties <module>**

where <module> can be one of the following values:

- • **security** to migrate only the security tables (only migrate these tables if you did not recreate the security setup in "Step 7: Configure the 4.5.2 Target Database" earlier)
- • **common** to migrate only the common tables specified in the properties file
- • **both** to migrate both security tables and the common tables specified in the properties file

**6** Review the log file created for this step. The log file is created in the folder from which you run the Java command, and is named **dm<date_time>.log**, where <date_time> is the date and time the log file was created.

**7**  Review the exception table *dm_aux_except* to see if there were any errors in processing the data.  If the table lists errors, fix the bad data and run step 5 again.

**8**  If you are running this process in steps, repeat steps 3 through 7 for each step until you have migrated all necessary common and security tables.

**9**  Continue to "Step 7:  Migrate Auxiliary Tables".

# Step 7:  Migrate Auxiliary Tables

Auxiliary tables include miscellaneous tables that are populated through the e*Index Administrator (previously e*Index Data Dictionary), such as the zip code table, predefined messages table, facility table, and so on.  They also include some miscellaneous tables such as the comments table and the user audit table.  As with the common tables, you can specify which tables to migrate.

## ▶ To migrate auxiliary tables

Before you begin:

  ✓  Complete "Step 6:  Migrate Security and Common Tables"

**1**  Navigate to the migration package home directory, and then to **\config**.

**2**  Open the file **AuxMigration.properties** in any text editor.

**3**  This file lists all of the 4.1.1 auxiliary tables that can be migrated.  To specify that a table be migrated, set the value to **true**.  To specify that a table not be migrated, set the value to **false**.

---

*Important!*

  ■  *You can perform this portion of the migration in steps, migrating only a few tables at a time, or you can migrate all tables at once.  You should not migrate any tables in which you have not added any custom information or modified existing records.*

  ■  *Unless you have made several customizations to the* ui_message *table, SeeBeyond recommends that you do not migrate this table due to the changes in terminology between version 4.1.1 and 4.5.2.*

  ■  *When you migrate the* ui_comments *table, the old terminology is not updated with the new terms.  If you do not migrate this table, all comments associated with member records will be lost.*

  ■  *If you did not customize* ui_control *and* ui_facility *earlier in "Step 7: Configure the 4.5.2 Target Database" (page 5-30), SeeBeyond highly recommends you migrate them.  We also recommend migrating* ui_comments *and* ui_audit *(unless you archive the table instead).*

---

**4**  Save and close **AuxMigration.properties**.

**5**  Run the following Java command:

> **java com.stc.eIndex.migration.Dm41ConvertAux**
> **Migration.properties**

**6**  Review the log file created for this step.  The log file is created in the folder from which you ran the Java command, and is named **dm<date_time>.log**, where <date_time> is the date and time the log file was created.

**7**  Review the exception table *dm_aux_except* to see if there were any errors in processing the data.  If the table lists errors, fix the errors and run step 5 again.

**8**  If you are running this process in steps, repeat steps 3 through 7 for each table until you have migrated all necessary auxiliary tables.

**9**  Continue to "Step 8:  Migrate Person Data".

# Step 8:  Migrate Person Data

This step of the migration process migrates all the person data, including history, alias, and local ID records, from the source database to the target database.  This step may take several days depending on the size of your database.  As with the previous tables, you can migrate person data in steps by specifying a date range for the records to be migrated.

## ▶ To migrate person data

Before you begin:

✓  Complete "Step 7:  Migrate Auxiliary Tables"

✓  Make sure that the COUNTRY control key in the 4.5.2 target database is set correctly (**USA** for installations in the United States and **AUS** for installations in Australia)

✓  Make sure that all facilities used in the 4.1.1 data have been migrated or defined in the 4.5.2 database.

---

*Tip:  To check the facilities, open a SQLPlus command for the 4.1.1 production database and type* **select unique facility from ui_local_id;**. *Then open a SQLPlus command for the 4.5.2 target database and type* **select facility_code from ui_facility;**.  *The resulting lists should be identical except the list for the 4.1.1 database will contain "**OLD #**".*

---

**1**  Run the following Java command to fix date issues with local ID records populated by the initial load procedure:

> **java com.stc.eIndex.migration.Dm41FixLocalId**
> **Migration.properties**

**2**  Review the log file created for this step.  The log file is created in the folder from which you run the Java command, and is named **dm<date_time>.log**, where <date_time> is the date and time the log file was created.

**3**  Navigate to the home directory and then to **\config**, and then open the file **Migration.properties**.

**4**  Modify the **StartDate** and **EndDate** variables so the date range only includes the dates you want to process.

> *Important!*  *Remember that the data must be processed in chronological order, so the **StartDate** should be the earliest create date in the e\*Index database for the first migration run.*

**5**  Run the following Java command:

> **java com.stc.eIndex.migration.Dm41ConvertHistory**
> **Migration.properties**

**6**  When the process is done running, review the log file created for this step.  The log file is created in the folder from which you run the Java command, and is named **dm<date_time>.log**, where <date_time> is the date and time the log file was created.

**7**  Do one of the following:

*If you are running the data migration in multiple runs, using the start and end dates to define each run,* continue to "Step 9: Review and Reprocess Exception Records", and then repeat steps 4 through 7 above, performing "Step 9: Review and Reprocess Exception Records" after each run.

*If you are running the data migration in a single run,* continue to "Step 9: Review and Reprocess Exception Records".

## Step 9: Review and Reprocess Exception Records

Any records that could not be processed during a given run of the person history migration are written to the exception tables in the source 4.1.1 database.  You can fix the errors and then reprocess the records. Check these tables frequently during the migration.

### ▶ To review and reprocess exception records

Before you begin:

✓ Complete "Step 8:  Migrate Person Data"

**1** Connect to the e*Index 4.1.1 source database using any SQL editor.

**2** Using SQL, view the data in the tables *dm_except_header*, *dm_except_detail*, and *dm_except_def*.  A sample select statement for viewing this information is provided below.

```
SELECT h.u_id, h.event_code, h.update_date, h.update_time,
   d.comments, def.descr, d.table_name, d.key_value,
   d.aux_id
FROM ui.dm_except_header h, ui.dm_except_detail d,
   ui.dm_except_def def
WHERE h.dm_except_header_id = d.dm_except_header_id
AND d.dm_except_def_id = def.dm_except_def_id
ORDER by h.u_id, h.dm_except_header_id;
```

*Tip:  You can customize the above script so it only displays the information you want to see.*

**3** Review the resulting information for any records that could not be migrated.

**4** Once you determine the source of the error, fix the data making sure not to modify any of the items listed under "Correcting Exception Records" on page 5-13.

*Tip:  Once a UID produces an error, all history records associated with that UID will also produce errors.  Often, you only need to fix one data error to be able to reprocess all of the error records associated with one UID.*

**5** For any records you fixed, change the value of the **reprocess** column in *dm_except_header* to **Y** to indicate that the record should be reprocessed. Remember to commit the change.

**6** Open the **Migration.properties** file in any text editor and change the value of the **ProcessExceptions** variable to **Y**.

**7** Save and close **Migration.properties**.

**8** Run the following Java command.

**java com.stc.eIndex.migration.Dm41ConvertHistory Migration.properties**

**9**   Review the exception tables once again to be sure that all records you wanted to reprocessed were processed correctly.  The **resolved** column changes to **Y** to indicate that the record was reprocessed successfully.

**10**   When you are finished reprocessing exception records:

- Change the value of the ProcessExceptions variable back to **N** in **Migration.properties**.

- Change the value of the **reprocess** column in *dm_except_header* back to **N** so the records do not get reprocessed again.

**11**   Repeat "Step 8:  Migrate Person Data" and "Step 9: Review and Reprocess Exception Records" for each phase of the person history data migration.  When you have processed all person history data, continue to "Step 10: Migrate Assumed Match Data"

## Step 10:  Migrate Assumed Match Data

The final step in migrating data is to migrate the assumed match information from the production database to the new database.  This step is only required if you want to maintain a history of assumed match information.  If you choose not to perform this step, all assumed match information will be lost.  Note that you must migrate all person data before performing this step; migrating the assumed match data relies on the transaction numbers created during the person data migration.

▶ **To migrate assumed match data**

Before you begin:

- ✓ Complete "Step 9: Review and Reprocess Exception Records"

**1**   Run the following Java command:

> **java com.stc.eIndex.migration.Dm41ConvertAssumedMatch Migration.properties**

**2**   Review the log file created for this step.  The log file is created in the folder from which you run the Java command, and is named **dm<date_time>.log**, where <date_time> is the date and time the log file was created.

**3**   Continue to "Step 11: Review the Migration Process"

## Step 11: Review the Migration Process

Once the migration is complete, you should do some comparisons between your source and the 4.5.2 target database to verify that the data was transferred correctly.  Some checks you can perform include the following:

- Verify that the *ui_person* tables in both the source and target databases contain the same number of records.

- Spot check individual person records by comparing the records using either the e*Index GUIs or SQL statements. Look for common table fields (such as religion, language, ethnicity, and so on), that display an identification code rather than a description. This indicates that you need to modify your common table data to accommodate the identification code.

- Review audit trails for a random sampling of UIDs to verify the information. Note that the audit trail functionality is more robust in version 4.5.2. You may have fewer audit trail records in version 4.5.2, but they provide a more accurate representation of a record's history.

- Verify that the *ui_person* tables in both the source and target databases contain the same number of records.

- Verify that the *ui_local_id* tables in both the source and target databases contain the same number of records.

- Verify that the *ui_alias* tables in both the source and target databases contain the same number of records.

- Verify that the number of telephone numbers stored in version 4.1.1 *ui_person* matches the number of records in version 4.5.2 *ui_phone*.

- Spot check individual records in *ui_person* for accuracy.

## Step 12:  Enable 4.5.2 Database Triggers

Earlier in the migration process, you disabled the triggers in the 4.5.2 target database. Before processing live data, you need to re-enable those triggers to process records through the new database.

### ▶ To enable 4.5.2 database triggers

Before you begin:

- ✓ Complete "Step 11: Review the Migration Process"

**1** From a Unix or MS-DOS command line, navigate to the home directory of the migration package, and then to **\db\oracle**.

**2** At the command prompt, type the following command:

   **sqlplus <un>/<up>@<db_name> @enable_triggers**

where

- <un> is your login ID for the e*Index 4.5.2 target database
- <up> is the password for the given login ID

- <db_name> is the Oracle TNS name of the e*Index 4.5.2 target database

**3** Continue to "Step 13: Load the Production Vality Rule Set".

## Step 13: Load the Production Vality Rule Set

Before you can bring the 4.5.2 database live, you need to load the Vality rule set files that you want to use into the production database. Chapter 6 of the *e*Index Administrator User's Guide* provides the background information and instructions you need to load the Vality information in the database. Before you begin, make sure you have the most current copy of the rule set files.

## Step 14: Assign a Region to Each System (optional)

If you installed region-specific security in your 4.5.2 target database, you need to assign a region to each system in the database before you can access any member information. To assign a region for each system, follow the instructions provided under "Assigning a System to a New Region" in chapter 4 of the *e*Index Administrator User's Guide*.

## Step 15: Update the Security Configuration

If you recreated your security setup when you configured your e*Index 4.5.2 database, you can skip this step. If you are using region-specific security, you may want to verify that the users defined in security are assigned regions.

e*Index Security was redesigned between versions 4.1.1 and 4.5.2. If you migrated the security tables instead of recreating the security setup, you may need to update the security configuration and reassign some access permissions. If you have Group Supervisors that require Administrator access, you need to recreate their profiles. For more information about the updates you may need to make, see "e*Index Security Considerations" in chapter 2 of this guide. Optionally, you can set up Event Notification at this time as well. Note that Event Notification requires the e-mail e*Way in order to transmit Event information.

## Step 16: Catch up from e*Gate Queue

Once you have reviewed the migration process, updated the security configuration, and made any necessary adjustments to the database, you can bring the 4.5.2 e*Index Schema live and start processing the queued data and live feeds into the e*Index database. Monitor this process closely in the beginning to ensure that data is being processed correctly.
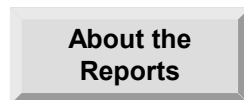
# Upgrading the Reports
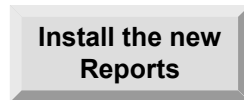
## About this Chapter

### Overview

This chapter presents the background information and the step-by-step instructions you need to upgrade the e*Index reports from version 4.1.1 to 4.5.2.

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

| | |
|---|---|
| **About the Reports** | Learn about the standard reports provided with e*Index |
| **Install the new Reports** | Learn how to install the new reports for version 4.5.2 |

# What's Inside

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

# Learning About the e\*Index Reports

## Overview

This section of the chapter provides background information about the e\*Index reports and the procedure for upgrading them.  It also includes summary information about the changes made to the reports for version 4.5.2.  For detailed information about these changes, see the *e\*Index 4.1.1 to 4.5.2 Release Bulletin.*

## About e\*Index Reports

SeeBeyond provides a standard set of database reports based on the PL/SQL query language for an e\*Index database on Oracle.  The reports are run daily, and report on the day-to-day changes to the database.  These reports tell you whether any potential duplicates were found, whether merges or unmerges took place, whether duplicate SSNs were found, and so on.  If you have a working knowledge of PL/SQL, you can customize these reports or create new reports to fit your reporting requirements.   You can also create reports using a standard report writer such as Crystal Reports.

## Getting Started

Before beginning, you should know where your current production reports are located, and how they have been customized.  In addition, any custom reports you created will need to be updated for the 4.5.2 database structure.  For information about the database differences between e\*Index 4.1.1 and 4.5.2, see "e\*Index Database Enhancements" in the *e\*Index 4.1.1 to 4.5.2 Release Bulletin*.

## Report Upgrade Process

Installing the updated reports is a simple procedure using the e\*Index 4.5.2 InstallShield.  Once you install the files, you need to customize them for your processing requirements and then copy them into your production environment.  Make sure to also upgrade any reports that you created for e\*Index that are not provided standard.

## 4.1.1 to 4.5.2 Reporting Differences

The primary differences between the reports for 4.1.1 to 4.5.2 are changes to accommodate a new date format and the new structure of the e\*Index database.  May of the new reports use a new table, *ui_transaction*, to obtain data to populate the reports.  In addition, some of the reports were reformatted to print more clearly.  For information about report differences, see "e\*Index Report Enhancements" in the *e\*Index 4.1.1 to 4.5.2 Release Bulletin*.

# Performing the Upgrade

## Overview

To upgrade the standard reports to e*Index 4.5.2, you must complete the following steps:

- Step 1: Back up the Production Reports
- Step 2: Install the Report Files
- Step 3: Customize the 4.5.2 Reports
- Step 4: Copy the Files to Report Home Directory

## Step 1: Back up the Production Reports

Before making any changes to your reporting environment, make backup copies of your production report files, including any new reports you created for e*Index.  After you install the updated reports, you can refer to your backup reports to see what customizations need to be made to the new report files and which custom reports you need to update.

## Step 2: Install the Report Files

Installing the report files is very similar to the process you followed to install the e*Index schema files.

### ▶ To install the database and report files

Before you begin:

- ✓ Complete "Step 1: Back up the Production Reports"
- ✓ Make sure the e*Index installation CD-ROM is inserted into the CD-ROM drive of your computer

**1**  Follow steps 1 through 7 under "Step 6: Install the e*Index Schema Files" in Chapter 3 of this guide.  The Select Components window should now be visible.

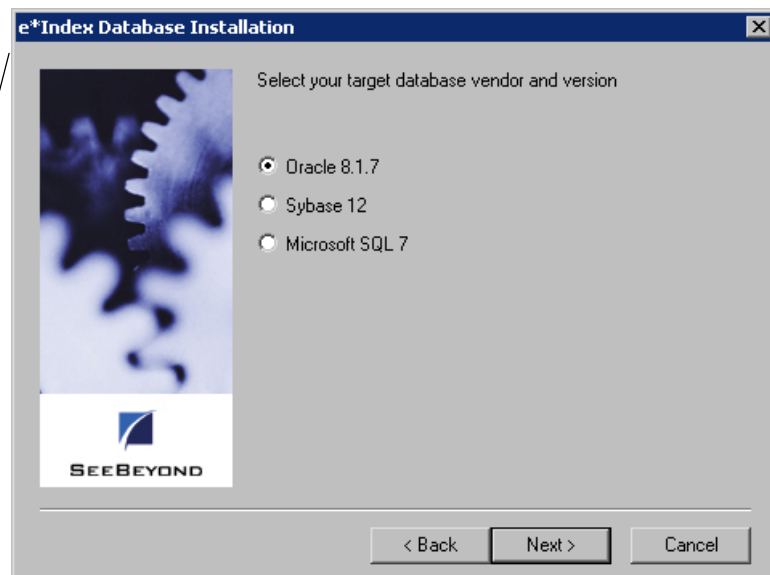**2**  In the Components box, select the check box next to **Database Scripts**.

Select the
components you
want to install on
the Select
Components
window
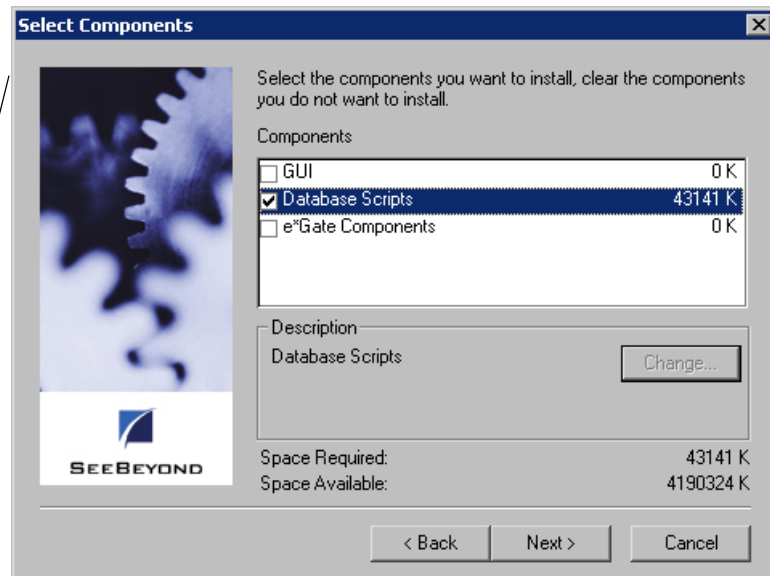
**3**  Click **Next**.  The e*Index Database Installation window appears.

*Next button*
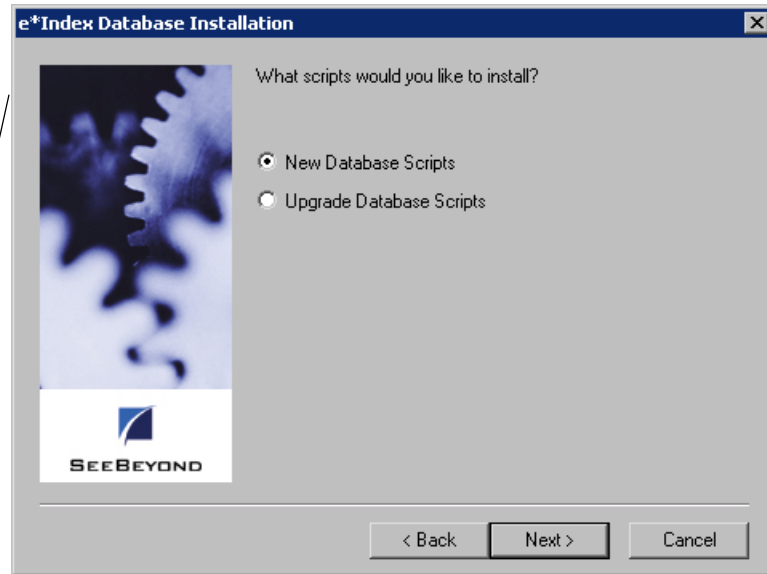
On the e*Index
Database
Installation
window, specify
Oracle 8.1.7

**4**  Select **Oracle 8.1.7**, and then click **Next**.  The second e*Index Database
    Installation window appears.

*Next button*

On the next e*Index
Database Installation
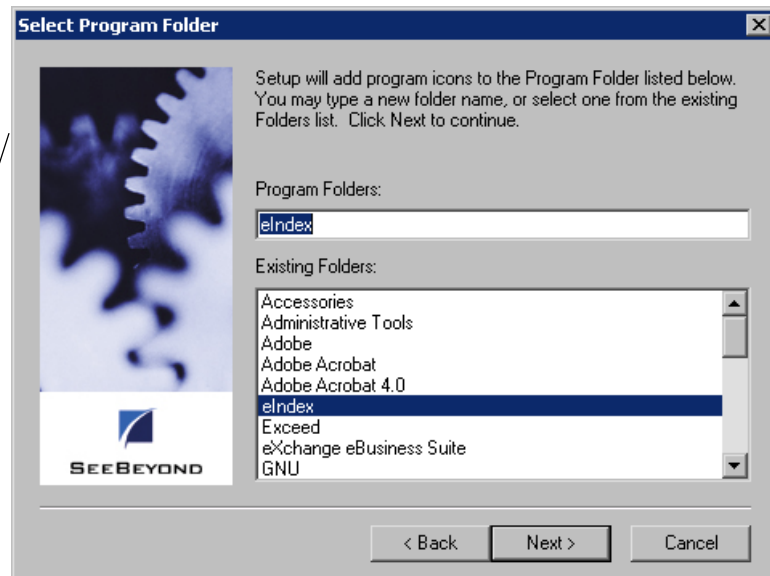window, specify
either option to
install the report files

**5**    Leave this window at it's default setting, and click **Next**.  The final
       e*Index Database Installation window appears.

*Next button*

On the next e*Index
Database Installation
window, specify that
you want to install
reports
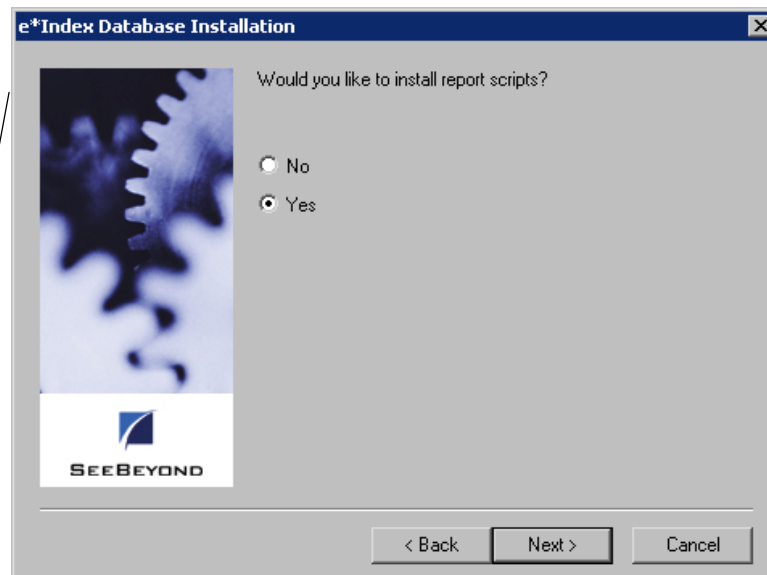
**6**    Select **Yes** to install the report files, and then click **Next**.  The Select
       Program Folder window appears.

*Next button*

Specify the folder in which to install the program icons on the Select Program Folder window

**7** Enter the name of the folder into which you want to install the program icons or accept the default name. Then click **Next**. The Check Setup Information window appears.

Next button

Verify the information you specified on the Check Setup Information window
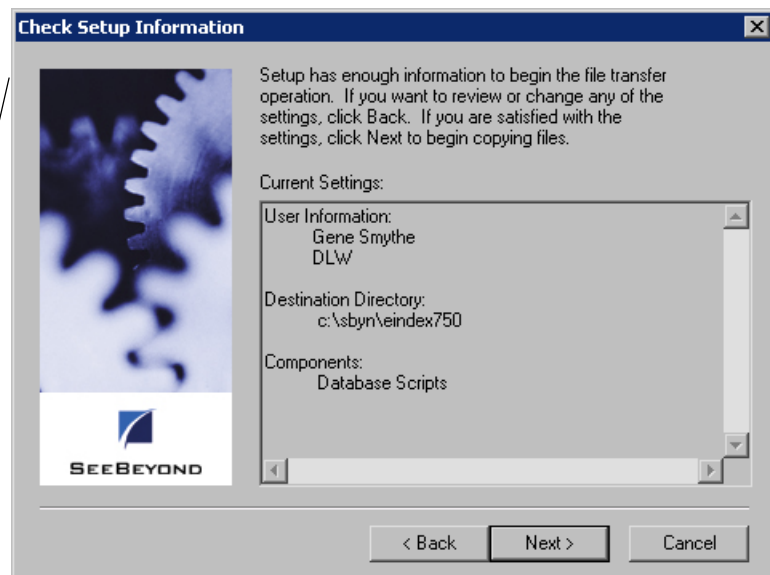
**8** Verify the information you specified, and do one of the following:

Back button

*To change any of the options you selected*, click **Back**, and make the necessary changes.

Next button

*To continue with the installation*, click **Next**. When all files are installed, the Setup Complete window appears.

The Setup
Complete window
indicates that the
database files are
installed
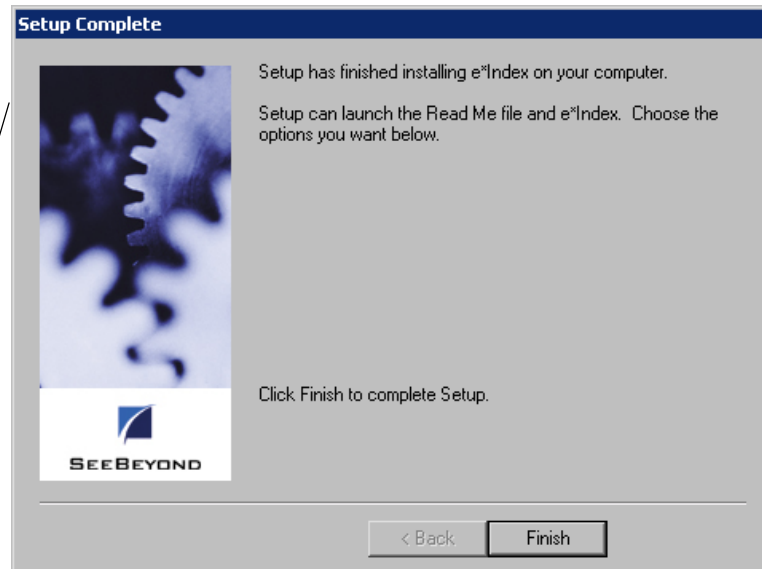
**9** Click **Finish** to complete the setup process.

*Finish button*

**10** To view the report files that were installed:

- Open Windows Explorer and navigate to the path you specified for the installation files.

- Open the **\server\UIreports** subdirectory.  This directory contains a subdirectory named **Production**, which contains all the report files.

**11** Continue to "Step 3: Customize the 4.5.2 Reports".

## Step 3: Customize the 4.5.2 Reports

The report files were restructured between version 4.1.1 and 4.5.2 to accommodate the many differences in the database structure.  If you customized the e*Index reports for 4.1.1, you will need to make similar customizations to the 4.5.2 reports.  For more information about the 4.5.2 reports, see the *Working with Reports for e*Index Global Identifier* for version 4.5.2.  For more information about changes to the reports and database structure, see the *e*Index 4.1.1 to 4.5.2 Release Bulletin*.

## Step 4: Move the Files to the Report Home Directory

If you did not install the report files in the location from which you want to run the reports, move the files to the report home directory.  Typically, SeeBeyond recommends placing these files in the database home directory so they can be easily located.  To move the report files, move all of the files located in the installation path in **\server\Uireports\Production** into the reports home directory.

# Administrative Table Data Mapping Chart

## About this Appendix

This appendix provides a chart that illustrates how data is processed from the 4.1.1 administrative tables to the 4.5.2 administrative tables.  In e*Index 4.5.2, much of the data entered through the e*Index Administrator has been condensed into one large table named *stc_common_detail*.  The header table for *stc_common_detail* is *stc_common_header*.  The chart on the following pages provides information about how the tables in version 4.1.1 are mapped to the new table structure in version 4.5.2.  It also provides suggestions on whether to truncate the information in the 4.5.2 tables before migrating, whether to customize the 4.5.2 data before migrating, and whether migrating specific tables is recommended.

The common header ID value for each data type in *stc_common_header* is provided in case you want to remove the 4.5.2 default data before migrating. The SQL statement for deleting common table data is provided under "Step 8: Remove Extraneous Code Table Data" in chapter 5, and requires that you know the common header ID of the data type you want to remove.

## Administrative Table Data Mapping Chart

| Data Type | 4.1.1 Table Name | 4.5.2 Table Name | 4.5.2 Common Header ID | Migration Recommended | Clear 4.5.2 Data Before Migrating | Customize 4.5.2 Data Before Migrating |
|---|---|---|---|---|---|---|
| Address Types | NA | stc_common_detail stc_common_header | 1 | NA | No | Yes |
| Application Messages | ui_message | ui_message | NA | No | Yes if migrating table | Not if migrating table |
| Citizenships | NA | stc_common_detail stc_common_header | 2 | NA | No | Yes |
| Countries | ui_country | stc_common_detail stc_common_header | 3 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| Departments | ui_dept | stc_common_detail stc_common_header | 4 | Yes | No | Not if migrating table |
| Districts of Residence | | stc_common_detail stc_common_header | 5 | NA | No | Yes |
| Driver License Issuers | NA | stc_common_detail stc_common_header | 6 | NA | No | Yes |
| Ethnicities | ui_ethnic | stc_common_detail stc_common_header | 7 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| Events | ui_event | stc_common_detail stc_common_header | 8 | Yes | No | Not if migrating table |
| Event Notifications | NA | stc_common_detail stc_common_header | 9 | NA | No | Yes |
| Genders | ui_sex | stc_common_detail stc_common_header | 10 | Only if table has been modified | Yes if migrating table | Not if migrating table |

| Data Type | 4.1.1 Table Name | 4.5.2 Table Name | 4.5.2 Common Header ID | Migration Recommended | Clear 4.5.2 Data Before Migrating | Customize 4.5.2 Data Before Migrating |
|---|---|---|---|---|---|---|
| Languages | ui_lang | stc_common_detail stc_common_header | 11 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| Marital Statuses | ui_mstatus | stc_common_detail stc_common_header | 12 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| Nationalities | NA | stc_common_detail stc_common_header | 13 | NA | No | Yes |
| Person Categories | NA | stc_common_detail stc_common_header | 14 | NA | No | Yes |
| Person Status | ui_status | stc_common_detail stc_common_header | 19 | Only if table has been modified | No | Not if migrating table |
| Phone Types | NA | stc_common_detail stc_common_header | 15 | NA | No | Yes |
| Pre-defined Messages | ui_canned_msg | ui_canned_msg | NA | Yes | No | Not if migrating table |
| Races | ui_race | stc_common_detail stc_common_header | 16 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| Regions | NA | stc_common_detail stc_common_header | 20 | NA | No | Only if using region-specific security |
| Religions | ui_religion | stc_common_detail stc_common_header | 17 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| State Codes | ui_state??? | stc_common_detail stc_common_header | 18 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| Suffixes | ui_suffix | stc_common_detail stc_common_header | 21 | Only if table has been modified | Yes if migrating table | Not if migrating table |

| Data Type | 4.1.1 Table Name | 4.5.2 Table Name | 4.5.2 Common Header ID | Migration Recommended | Clear 4.5.2 Data Before Migrating | Customize 4.5.2 Data Before Migrating |
|---|---|---|---|---|---|---|
| Sources (systems) | ui_system | stc_common_detail stc_common_header | 22 | Yes | No | Not if migrating table |
| Systems (facilities) | ui_facility | ui_facility | NA | Yes | Yes if migrating table | Not if migrating table |
| Titles | ui_title | stc_common_detail stc_common_header | 23 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| Veteran Statuses | ui_veteran | stc_common_detail stc_common_header | 24 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| VIP Flags | ui_vip | stc_common_detail stc_common_header | 25 | Only if table has been modified | Yes if migrating table | Not if migrating table |
| Zip Codes | ui_zip | ui_zip | NA | Only if table has been modified | Yes if migrating table | Not if migrating table |