SeeBeyond ICAN Suite

eIndex Global Identifier Composite Application Upgrade Guide

Upgrading from Release 4.5.3 to Release 5.0.4

SeeBeyond Proprietary and Confidential

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

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

© 2004 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

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

Version 20040603101111.

Contents

Chapter 1

Introduction 5

Document Purpose and Scope 5
Intended Audience 5
Using this Guide 5
Document Organization 6

Writing Conventions 6
Special Notation Conventions 6
Mouse Conventions 7

Supporting Documents 8

Online Documents 8

SeeBeyond Web Site 8

Chapter 2

Upgrade Process Overview 9

Upgrade Overview 9
Upgrading the Schema to a Project 9
Upgrading the Outbound Queue 10
Upgrading the GUI 10
Upgrading to the EDM 10
Upgrading Security 10
Upgrading the Administrative Configuration 10
Upgrading the Reports 11
Upgrading the Java API for eIndex Active Integration 11
Upgrading the Database 12
Methodology 12
Migration Phases 12
How Person Data is Migrated 12
Obsolete Tables 13
Database Customizations 13
About eGate Queuing 14

eIndex 5.0.4 Oracle Database Model 14

Chapter 3

Installation 19

System Requirements 19

Installing the Upgrade Files 20

Step 1: Install eIndex 5.0.4 20
Step 2: Download the Upgrade Files 20
Step 3: Set the Environment Variables 21

Chapter 4

Updating the eIndex Project 22

General Updates 22

eIndex Configuration 22
Defining a Data Connection 22
Customizing Connectivity 22
Defining the Environment and Deployment Profile 23
Setting up Security 23

Updating the Unmerge Policy 23

Chapter 5

Migrating the Database 25

Learning About the Properties Files 25

migration.properties 25
source.properties 25
target.properties 26
log4j.properties 27

Setting up the Databases 28

Step 1: Create and Configure the Source Database 28
Step 2: Create the eIndex 5.0.4 Target Database 28
Step 3: Begin Queuing 4.5.3 and 5.0.4 Daily Transactions 29
Step 4: Back up the Production Database 29
Step 5: Export and Import the Production Data 29
Step 6: Restart the 4.5.3 eWays 29

Migrating the Database 29

Step 1: Modify the Properties Files 30
Step 2: Perform the Initial Data Transfer 30
Step 3: Migrate History Data 31
Step 4: Update the Sequence Table 32
Step 5: Review the Migration Process 32
Step 6: Catch up from eGate Queue 32

Index 33

Chapter 1

Introduction

This guide provides comprehensive information on upgrading the SeeBeyond® e*Index™ Global Identifier (e*Index) version 4.5.3 to eIndex™ Global Identifier Composite Application (eIndex) version 5.0.4. This guide primarily focuses on upgrading the database, but also includes information on upgrading the schema, GUI, and the Java APIs for Active Integration. This chapter provides an overview of this guide and the conventions used throughout, as well as a list of supporting documents and information about using this guide.

1.1 Document Purpose and Scope

This guide provides step-by-step instructions for upgrading eIndex. It includes navigational information, functional instructions, and background information where required.

This guide does not include information or instructions on using the EDM or working with Enterprise Designer or eGate Integrator components. These topics are covered in the appropriate user guide (for more information, see **"Document Purpose and Scope" on page 5**).

1.1.1 Intended Audience

Any user who upgrades any component of eIndex should read this guide. A thorough knowledge of eIndex is not needed to understand this guide. It is presumed that the reader of this guide is familiar with the eGate environment and GUIs, eGate Projects, Oracle database administration, and the operating system(s) on which eGate and the eIndex database run. Familiarity with XML documents, the SQL scripting language, and Java is helpful for the upgrade.

The intended reader must have a good working knowledge of his or her company's current business processes and information system (IS) setup, as well as the current e*Index implementation.

1.1.2 Using this Guide

For best results, skim through the guide to familiarize yourself with the locations of essential information you need. The beginning of each chapter provides introductory information on the topics covered in that chapter. This introductory material contains

background and explanatory information you may need to understand before moving into the more detailed information later in the chapter.

1.1.3 Document Organization

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

- **Chapter 1 "Introduction"** gives a general preview of this document—its purpose, scope, and organization—and provides sources of additional information.
- **Chapter 2 "Upgrade Process Overview"** gives a general summary of the upgrade process, describes installation requirements, and includes information about the database upgrade.
- **Chapter 3 "Installation"** gives information and instructions for updating an active e*Index environment to version 5.0.4.
- **Chapter 4 "Updating the eIndex Project"** gives general information about the types of changes to make to the eIndex Project prior to migrating the database, and includes instructions on updating the unmerge policy.
- **Chapter 5 "Migrating the Database"** gives instructions for creating a new eIndex 5.0.4 database and migrating existing data from version 4.5.3 to the new database.

1.2 Writing Conventions

Before you start using this guide, it is important to understand the special notation and mouse conventions observed throughout this document.

1.2.1 Special Notation Conventions

The following special notation conventions are used in this document.

Table 1 Special Notation Conventions

Text	Convention	Example
Titles of publications	Title caps in *italic* font	*eIndex Global Identifier Composite Application User's Guide*
Button, Icon, Command, Function, and Menu Names	**Bold** text	▪ Click **OK** to save and close. ▪ From the **File** menu, select **Exit**.
Parameter, Variable, and Method Names	**Bold** text	▪ Use the **executeMatch()** method. ▪ Enter the **field-type** value.
Command Line Code and Code Samples	`Courier` font (variables are shown in ***bold italic***)	▪ `bootstrap -p ***password***` ▪ `<tag>Person</tag>`

Table 1 Special Notation Conventions (Continued)

Text	Convention	Example
Hypertext Links	**Blue** text	For more information, see **"Writing Conventions" on page 6**.
File Names and Paths	**Bold** text	To install eIndex, upload the **eIndex.sar** file.
Notes	***Bold Italic*** text	***Note:*** *If a toolbar button is dimmed, you cannot use it with the selected component.*

Additional Conventions

Windows Systems—The eIndex system is fully compliant with Windows 2000, Windows XP, and Windows 2003 platforms. When this document refers to Windows, such statements apply to all three Windows platforms.

UNIX Systems—This guide uses the backslash (\ ) as the separator within path names. If you are working on a UNIX system, please make the appropriate substitutions.

1.2.2 Mouse Conventions

You can use either a single-button mouse or a multiple-button mouse with eIndex. 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 and then 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 eIndex 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.

1.3 Supporting Documents

SeeBeyond has developed a suite of user's guides and related publications that are distributed in an electronic library. The following documents may provide information useful in creating your customized index. In addition, complete documentation of the eIndex Java API is provided in Javadoc format.

- *eIndex Enterprise Data Manager User's Guide*
- *eIndex Global Identifier Composite Application Configuration Guide*
- *eIndex Global Identifier Composite Application Reference Guide*
- *Implementing the SeeBeyond Match Engine with eIndex*
- *Implementing Ascential INTEGRITY with eIndex*
- *Working with Reports for eIndex*
- *eGate Integrator User's Guide*
- *eGate Integration System Administration Guide*
- *SeeBeyond ICAN Suite Deployment Guide*

1.4 Online Documents

The documentation for the SeeBeyond ICAN Suite is distributed as a collection of online documents. These documents are viewable with the Acrobat Reader application from Adobe Systems. Acrobat Reader can be downloaded from:

http://www.adobe.com

1.5 SeeBeyond Web Site

The SeeBeyond Web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

http://www.SeeBeyond.com

Chapter 2

Upgrade Process Overview

This chapter presents background information about the process for upgrading an e*Index 4.5.3 environment to eIndex 5.0.4. It also includes important information about the database upgrade process.

2.1 Upgrade Overview

Moving your current environment to version 5.0.4 requires upgrading several components. The following steps are required to complete the eIndex upgrade.

- **Upgrading the Schema to a Project** on page 9
- **Upgrading the Outbound Queue** on page 10
- **Upgrading the GUI** on page 10
- **Upgrading the Reports** on page 11
- **Upgrading the Java API for eIndex Active Integration** on page 11
- **Upgrading the Database** on page 12

Note: *This section provides overview information about upgrading the configuration of eIndex. For more specific information, refer to the* ***eIndex Global Identifier Composite Application User's Guide*** *and the* ***eIndex Global Identifier Composite Application Configuration Guide****.*

2.1.1 Upgrading the Schema to a Project

eIndex 5.0.4 provides a Project that processes data in the same manner as the e*Index 4.5.3 sample schema. To upgrade the schema, install eIndex in the Enterprise Designer, and then customize the new Project based on the customizations you made to the 4.5.3 schema. In eIndex 5.0.4, you will be using Java methods rather than Monk functions to customize how data is processed. For more information about the Java functions that are available to eGate Collaborations, see chapter 5 of the *eIndex Global Identifier Composite Application Reference Guide*. For information about general updates to make to the Project, see **Chapter 4**, **"Updating the eIndex Project"**.

2.1.2 Upgrading the Outbound Queue

In previous versions, when certain updates were made using the e*Index GUI, e*Index published messages to a database table (the "outbound queue") with the new information. These messages were removed from the outbound queue by the polling eWay, and forwarded to eGate for routing to external systems. eIndex 5.0.4 publishes all messages processed through eIndex directly to a JMS Topic for routing. To upgrade the outbound queue, you need to add a JMS Topic to the eIndex server Project. This topic can then be included in external system Projects, along with a Service to process the data to the external systems. The outbound queue in the e*Index 4.5.3 database must be empty before beginning the database portion of the upgrade.

2.1.3 Upgrading the GUI

e*Index 4.5.3 had three user interfaces: e*Index Global Identifier, e*Index Administrator, and e*Index Security. eIndex 5.0.4 retains only the global identifier GUI, which is now a Web-based browser called the *Enterprise Data Manager* (EDM). The functions previously performed using the e*Index Administrator are either obsolete or are now handled through the XML configuration files in the eIndex Project. Security functions are now handled through the Integration Server.

Upgrading to the EDM

The only step for upgrading the eIndex GUI is to ensure that the client workstation has network access to the Integration Server machine and has Internet Explorer 6.0 with SP1 or later. If the appearance of the e*Index Global Identifier 4.5.3 GUI was customized, these customizations must be made manually to the Enterprise Data Manager file in the eIndex Project. This includes showing and hiding fields, modifying labels, modifying field names, defining field formats, and so on.

Upgrading Security

Previously, e*Index handled security through its own security module, which restricted access at the user, function, and action levels. eIndex 5.0.4 relies on the SeeBeyond Integration Server to handle security. Any security information previously stored in e*Index must be transferred manually to the Integration Server. eIndex no longer supports the Event Notification function; however, this can be handled through eGate and the e-mail eWay.

Upgrading the Administrative Configuration

The e*Index Administrator allowed you to maintain several types of data stored in the database, including:

- Control key values
- Configurable queries
- Common table data
- INTEGRITY rule set information

Of these, only common table data is stored in the eIndex 5.0.4 database, and is the only component that is migrated automatically. The rest of the components must be configured in the configuration files of the eIndex Project.

Control Key Values

Most control keys have been removed as obsolete. Weight thresholds and other matching control keys, such as one exact match and duplicate check on update, are defined manually in the eIndex Matching Service (the Match Field configuration file) and must be configured in the eIndex Project.

Configurable Queries

The configurable queries define fields to block for matching, and the combinations of fields allowed for a phonetic search. Previously, the configurable queries were modified through e*Index Administrator, and the information was stored in the database. In version 5.0.4, the configurable blocking queries are customized in the Query Builder (the Candidate Select file), and the information is stored in that file. eIndex 5.0.4 does not provide the ability to include constants in the "where" clause of the SQL statements, but it provides greater flexibility in choosing fields on which to block.

Extensive searching was previously defined in the configurable query for record processing, and by a control key for GUI searches. In version 5.0.4, it is defined by the blocker query and the ALPHA-SEARCH query in the Query Manager. By default, the blocking query and phonetic search are set up for extensive searching, and you can modify the query configuration if needed.

Common Table Data

All common table data is automatically migrated into the eIndex 5.0.4 database.

INTEGRITY Rule Set Files

Rule set information is no longer stored in the database, making the Rule Set Maintenance function obsolete. Rule set files only need to be upgraded if they have been customized in version 4.5.3. To upgrade the rule set files, the customized 4.5.3 rule set files must be copied to the machine on which the integration server resides, and the path to the rule set files must be specified by the environment variable VTICFG.

2.1.4 Upgrading the Reports

eIndex 5.0.4 does not include the standard PL/SQL reports from version 4.5.3, but does provide the following reports through a Java-based report client: assumed match, unmerge, merge, potential duplicate, and deactivate. You can modify your existing reports to query the updated database. Both database are transaction-based, so the report structure is similar. For more information about the structure of the 5.0.4 database, see **"eIndex 5.0.4 Oracle Database Model" on page 14**.

2.1.5 Upgrading the Java API for eIndex Active Integration

e*Index 4.5.3 provides Java APIs that can be called by a third party to access information in the database. This API must be upgraded manually for version 5.0.4. In version 5.0.4, all eIndex functions are written in Java, and can be used to upgrade the 4.5.3 active integration code. Customers currently on 4.5.3 who use the functions

directly from the eGate Collaborations can upgrade by using the method OTDs rather than writing straight Java in the Collaborations.

2.1.6 Upgrading the Database

Upgrading the database involves migrating your production data from a copy of the 4.5.3 production database to a new 5.0.4 database. eIndex 5.0.4 includes several new tables, and makes several of the 4.5.3 tables obsolete.

Methodology

The migration process involves three databases: the production database, an e*Index 4.5.3 source database, and an eIndex 5.0.4 target database. To begin the upgrade process, data is exported from the production database and then imported into the 4.5.3 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 4.5.3 source database to the 5.0.4 target database. The 5.0.4 target database becomes the production database once the upgrade and testing are complete.

To ensure no information is lost during the upgrade, inbound messages are stored in the eGate 5.0.4 queue while the upgrade is performed. Once the export of the production database begins, no changes to person data can be made from the e*Index Global Identifier GUI.

Migration Phases

The migration of data from the 4.5.3 source database to the 5.0.4 target database is performed in three phases.

1. Migrate current person data and supporting data
2. Migrate transaction history data
3. Update the sequence table

How Person Data is Migrated

When you migrate person information from the 4.5.3 source database to the 5.0.4 target database, current person data is migrated first. This data is inserted into the system object tables and the single best record (SBR) tables. Table 2 lists the 4.5.3 to 5.0.4 mapping of tables containing person data.

Table 2 Table Mapping from Version 4.5.3 to 5.0.4

Data in this 4.5.3 table…	is migrated to this SBR table…	and to this system object table
ui_person	sbyn_personsbr	sbyn_person
ui_address	sbyn_addresssbr	sbyn_address
ui_alias	sbyn_aliassbr	sbyn_alias
ui_auxid	sbyn_auxidsbr	sbyn_auxid

Table 2 Table Mapping from Version 4.5.3 to 5.0.4 (Continued)

Data in this 4.5.3 table…	is migrated to this SBR table…	and to this system object table
ui_comment	sbyn_commentsbr	sbyn_comment
ui_phone	sbyn_phonesbr	sbyn_phone

History information is no longer stored as complete person records in the *_history database tables. This information is now stored in one table, sbyn_transaction, which stores a history of only the changes made in each transaction. This means that migrating transaction history data requires the program to translate each history record to a description of the changes that occurred for that transaction. The data migration process inserts each translation into a row in sbyn_transaction, with a complete description of the changes made.

Obsolete Tables

The structure of eIndex 5.0.4 simplifies the database, and several tables are now obsolete. Most of the information previously stored in these tables is now stored in the XML configuration files, which are read at runtime. The history tables have been consolidated into a single delta table, and the functions handled by both the security tables and the outbound event tables have been transferred to eGate. Rule set information is now only stored in the rule set files, and is no longer stored in the database as well. The following tables are obsolete:

- Rule set tables
- Configurable query tables
- Country configuration tables
- Display configuration tables
- History tables (ui_*_history)
- Case-insensitive search tables
- Security tables
- Outbound event tables
- ui-config
- ui-control
- ui_canned_msg
- ui_message
- ui_zip

Database Customizations

If you have customized your database by creating any of the following items, they will not be transferred to the new database during the migration process. To maintain the functionality of these items, you must add them to the new eIndex 5.0.4 database manually.

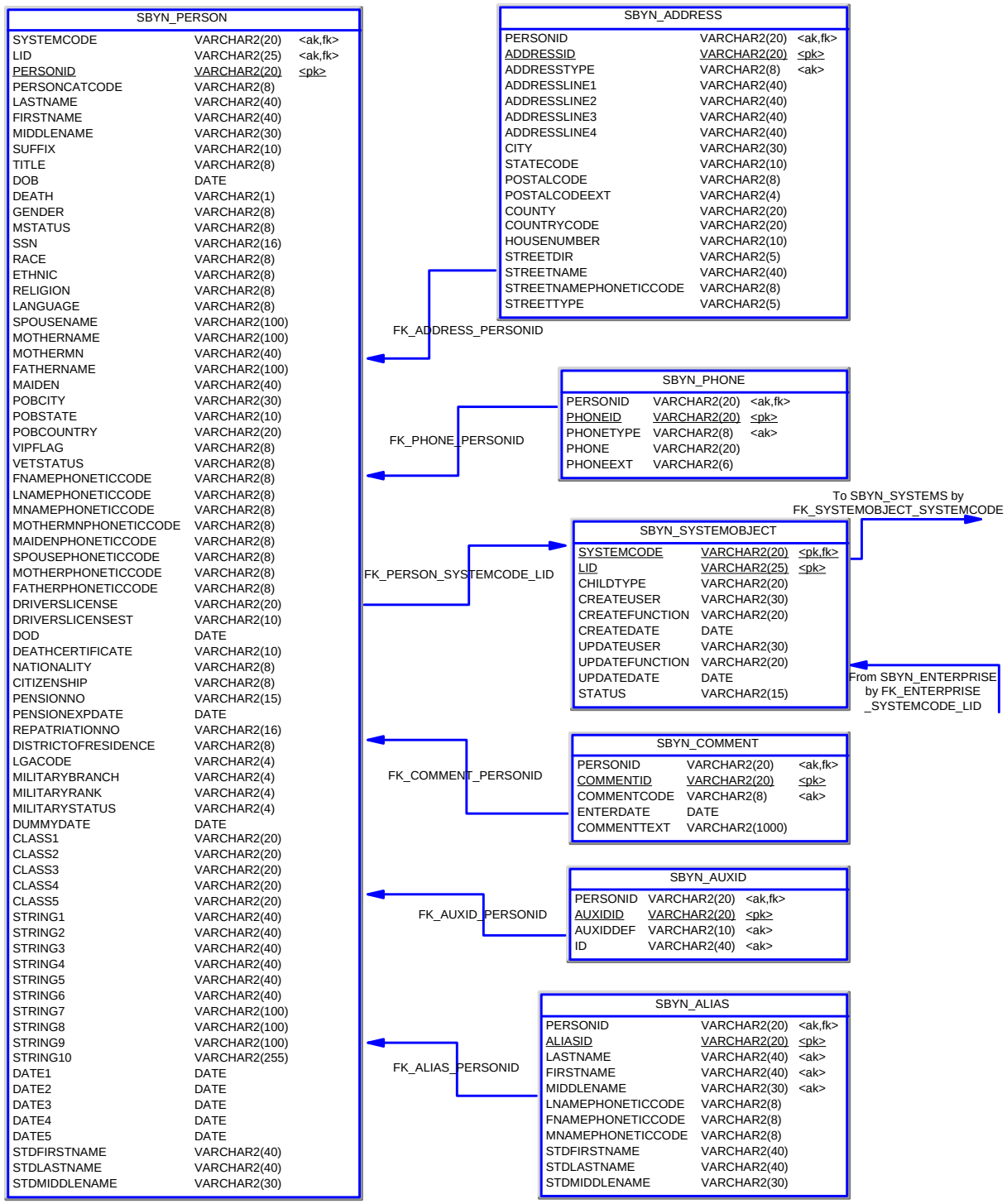
- New database tables
- New columns added to existing tables
- New stored procedures
- New triggers
- New views

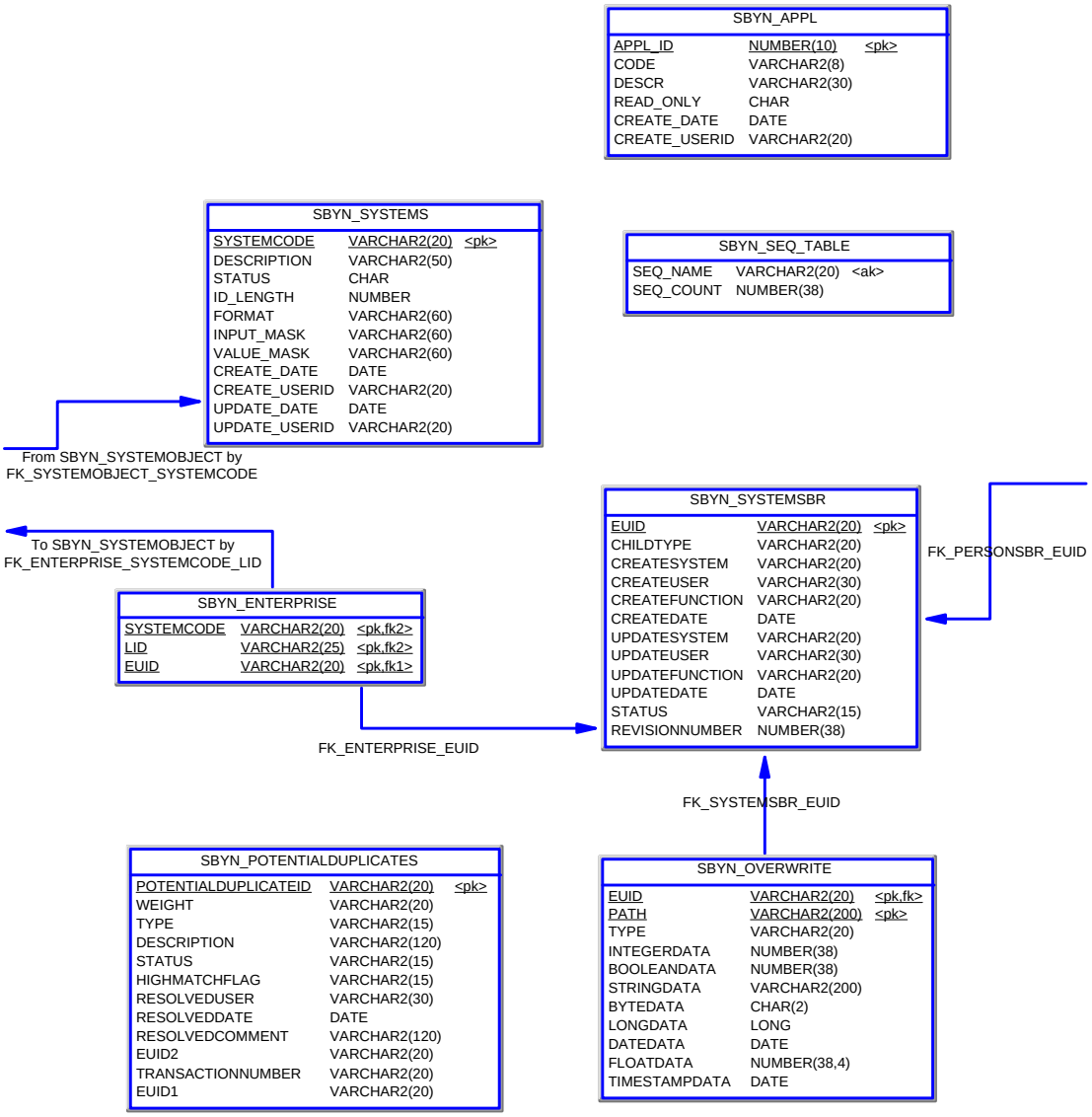
About eGate Queuing

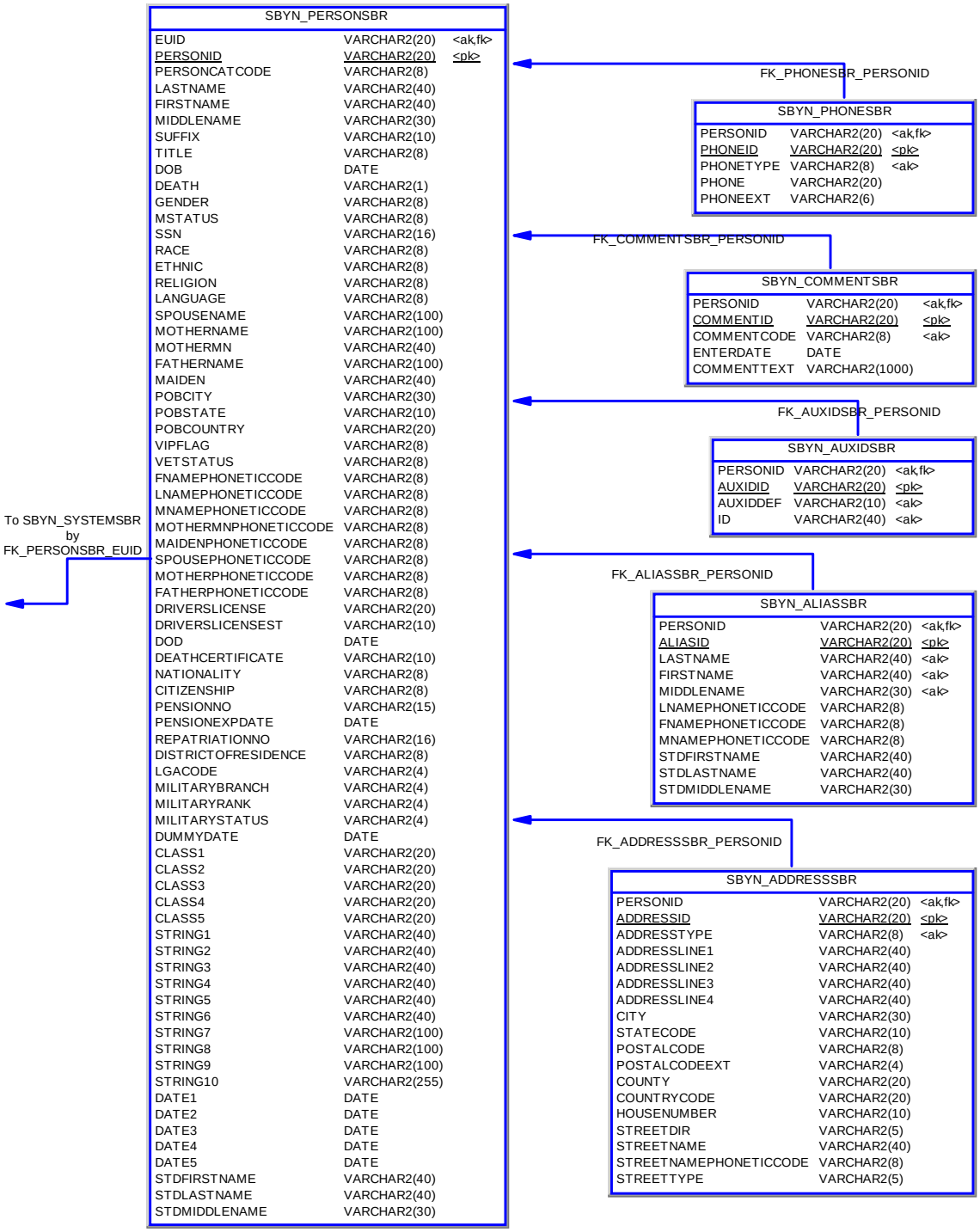
When you export your production database, queue any incoming transactions to the 4.5.3 schema to avoid losing any information. At the same time, you also begin queuing the incoming transactions to the 5.0.4 Project. During the migration process, you can continue to use the e*Index 4.5.3 production database to process daily transactions, but at the same time you must continue queuing those transactions in the 5.0.4 Project. When the 5.0.4 database is ready to move into production, you can start processing the stored Events from the 5.0.4 queue to bring the database current.

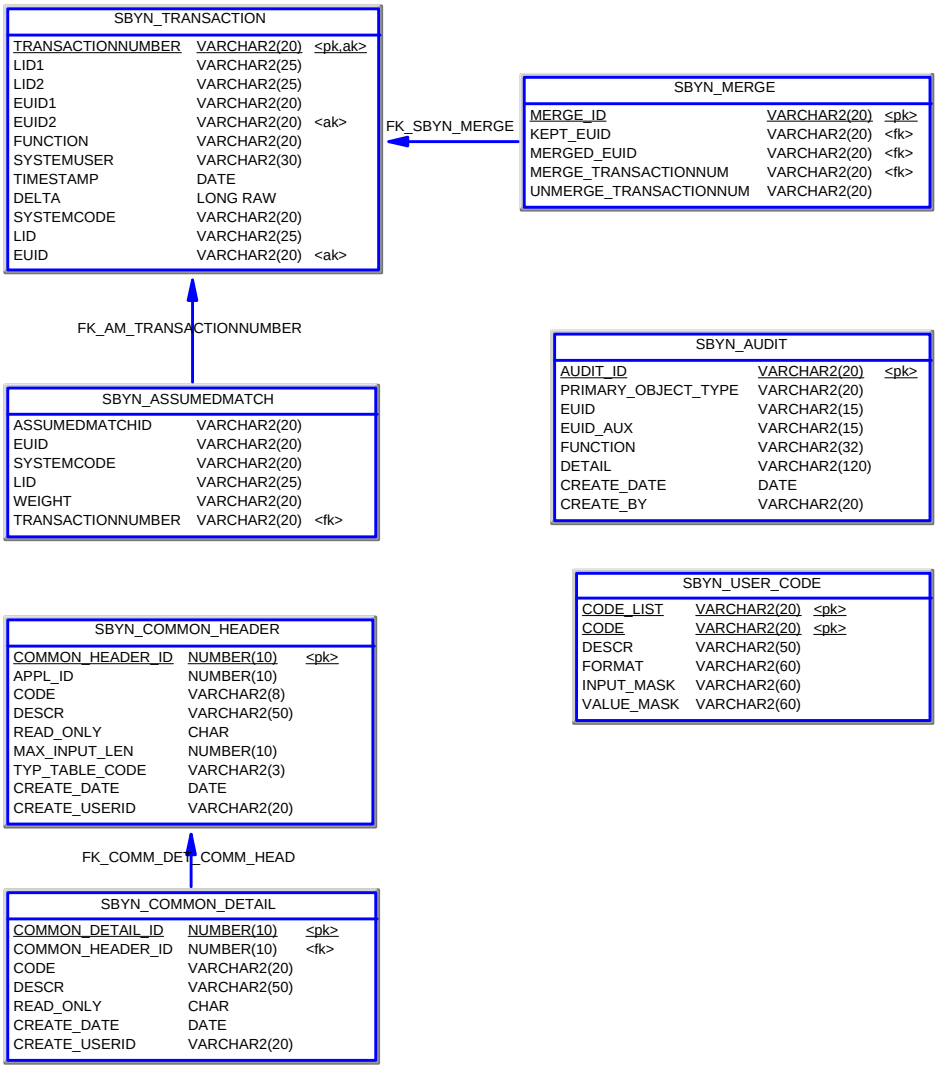
2.2 eIndex 5.0.4 Oracle Database Model

The diagrams on the following pages illustrate the table structure for a default implementation of eIndex 5.0.4 for Oracle. Your actual database may vary slightly from this model depending upon the release you have installed.


SBYN_PERSON
SYSTEMCODE VARCHAR2(20) <ak,fk>
LID VARCHAR2(25) <ak,fk>
PERSONID VARCHAR2(20) <pk>
PERSONCATCODE VARCHAR2(8)
LASTNAME VARCHAR2(40)
FIRSTNAME VARCHAR2(40)
MIDDLENAME VARCHAR2(30)
SUFFIX VARCHAR2(10)
TITLE VARCHAR2(8)
DOB DATE
DEATH VARCHAR2(1)
GENDER VARCHAR2(8)
MSTATUS VARCHAR2(8)
SSN VARCHAR2(16)
RACE VARCHAR2(8)
ETHNIC VARCHAR2(8)
RELIGION VARCHAR2(8)
LANGUAGE VARCHAR2(8)
SPOUSENAME VARCHAR2(100)
MOTHERNAME VARCHAR2(100)
MOTHERMN VARCHAR2(40)
FATHERNAME VARCHAR2(100)
MAIDEN VARCHAR2(40)
POBCITY VARCHAR2(30)
POBSTATE VARCHAR2(10)
POBCOUNTRY VARCHAR2(20)
VIPFLAG VARCHAR2(8)
VETSTATUS VARCHAR2(8)
FNAMEPHONETICCODE VARCHAR2(8)
LNAMEPHONETICCODE VARCHAR2(8)
MNAMEPHONETICCODE VARCHAR2(8)
MOTHERMNPHONETICCODE VARCHAR2(8)
MAIDENPHONETICCODE VARCHAR2(8)
SPOUSEPHONETICCODE VARCHAR2(8)
MOTHERPHONETICCODE VARCHAR2(8)
FATHERPHONETICCODE VARCHAR2(8)
DRIVERSLICENSE VARCHAR2(20)
DRIVERSLICENSEST VARCHAR2(10)
DOD DATE
DEATHCERTIFICATE VARCHAR2(10)
NATIONALITY VARCHAR2(8)
CITIZENSHIP VARCHAR2(8)
PENSIONNO VARCHAR2(15)
PENSIONEXPDATE DATE
REPATRIATIONNO VARCHAR2(16)
DISTRICTOFRESIDENCE VARCHAR2(8)
LGACODE VARCHAR2(4)
MILITARYBRANCH VARCHAR2(4)
MILITARYRANK VARCHAR2(4)
MILITARYSTATUS VARCHAR2(4)
DUMMYDATE DATE
CLASS1 VARCHAR2(20)
CLASS2 VARCHAR2(20)
CLASS3 VARCHAR2(20)
CLASS4 VARCHAR2(20)
CLASS5 VARCHAR2(20)
STRING1 VARCHAR2(40)
STRING2 VARCHAR2(40)
STRING3 VARCHAR2(40)
STRING4 VARCHAR2(40)
STRING5 VARCHAR2(40)
STRING6 VARCHAR2(40)
STRING7 VARCHAR2(100)
STRING8 VARCHAR2(100)
STRING9 VARCHAR2(100)
STRING10 VARCHAR2(255)
DATE1 DATE
DATE2 DATE
DATE3 DATE
DATE4 DATE
DATE5 DATE
STDFIRSTNAME VARCHAR2(40)
STDLASTNAME VARCHAR2(40)
STDMIDDLENAME VARCHAR2(30)
SBYN_ADDRESS
PERSONID VARCHAR2(20) <ak,fk>
ADDRESSID VARCHAR2(20) <pk>
ADDRESSTYPE VARCHAR2(8) <ak>
ADDRESSLINE1 VARCHAR2(40)
ADDRESSLINE2 VARCHAR2(40)
ADDRESSLINE3 VARCHAR2(40)
ADDRESSLINE4 VARCHAR2(40)
CITY VARCHAR2(30)
STATECODE VARCHAR2(10)
POSTALCODE VARCHAR2(8)
POSTALCODEEXT VARCHAR2(4)
COUNTY VARCHAR2(20)
COUNTRYCODE VARCHAR2(20)
HOUSENUMBER VARCHAR2(10)
STREETDIR VARCHAR2(5)
STREETNAME VARCHAR2(40)
STREETNAMEPHONETICCODE VARCHAR2(8)
STREETTYPE VARCHAR2(5)
FK_ADDRESS_PERSONID
SBYN_PHONE
PERSONID VARCHAR2(20) <ak,fk>
PHONEID VARCHAR2(20) <pk>
PHONETYPE VARCHAR2(8) <ak>
PHONE VARCHAR2(20)
PHONEEXT VARCHAR2(6)
FK_PHONE_PERSONID
To SBYN_SYSTEMS by FK_SYSTEMOBJECT_SYSTEMCODE
SBYN_SYSTEMOBJECT
SYSTEMCODE VARCHAR2(20) <pk,fk>
LID VARCHAR2(25) <pk>
CHILDTYPE VARCHAR2(20)
CREATEUSER VARCHAR2(30)
CREATEFUNCTION VARCHAR2(20)
CREATEDATE DATE
UPDATEUSER VARCHAR2(30)
UPDATEFUNCTION VARCHAR2(20)
UPDATEDATE DATE
STATUS VARCHAR2(15)
FK_PERSON_SYSTEMCODE_LID
From SBYN_ENTERPRISE by FK_ENTERPRISE_SYSTEMCODE_LID
SBYN_COMMENT
PERSONID VARCHAR2(20) <ak,fk>
COMMENTID VARCHAR2(20) <pk>
COMMENTCODE VARCHAR2(8) <ak>
ENTERDATE DATE
COMMENTTEXT VARCHAR2(1000)
FK_COMMENT_PERSONID
SBYN_AUXID
PERSONID VARCHAR2(20) <ak,fk>
AUXIDID VARCHAR2(20) <pk>
AUXIDDEF VARCHAR2(10) <ak>
ID VARCHAR2(40) <ak>
FK_AUXID_PERSONID
SBYN_ALIAS
PERSONID VARCHAR2(20) <ak,fk>
ALIASID VARCHAR2(20) <pk>
LASTNAME VARCHAR2(40) <ak>
FIRSTNAME VARCHAR2(40) <ak>
MIDDLENAME VARCHAR2(30) <ak>
LNAMEPHONETICCODE VARCHAR2(8)
FNAMEPHONETICCODE VARCHAR2(8)
MNAMEPHONETICCODE VARCHAR2(8)
STDFIRSTNAME VARCHAR2(40)
STDLASTNAME VARCHAR2(40)
STDMIDDLENAME VARCHAR2(30)
FK_ALIAS_PERSONID


SBYN_APPL
APPL_ID NUMBER(10) <pk>
CODE VARCHAR2(8)
DESCR VARCHAR2(30)
READ_ONLY CHAR
CREATE_DATE DATE
CREATE_USERID VARCHAR2(20)
SBYN_SYSTEMS
SYSTEMCODE VARCHAR2(20) <pk>
DESCRIPTION VARCHAR2(50)
STATUS CHAR
ID_LENGTH NUMBER
FORMAT VARCHAR2(60)
INPUT_MASK VARCHAR2(60)
VALUE_MASK VARCHAR2(60)
CREATE_DATE DATE
CREATE_USERID VARCHAR2(20)
UPDATE_DATE DATE
UPDATE_USERID VARCHAR2(20)
SBYN_SEQ_TABLE
SEQ_NAME VARCHAR2(20) <ak>
SEQ_COUNT NUMBER(38)
From SBYN_SYSTEMOBJECT by
FK_SYSTEMOBJECT_SYSTEMCODE
To SBYN_SYSTEMOBJECT by
FK_ENTERPRISE_SYSTEMCODE_LID
SBYN_ENTERPRISE
SYSTEMCODE VARCHAR2(20) <pk,fk2>
LID VARCHAR2(25) <pk,fk2>
EUID VARCHAR2(20) <pk,fk1>
SBYN_SYSTEMSBR
EUID VARCHAR2(20) <pk>
CHILDTYPE VARCHAR2(20)
CREATESYSTEM VARCHAR2(20)
CREATEUSER VARCHAR2(30)
CREATEFUNCTION VARCHAR2(20)
CREATEDATE DATE
UPDATESYSTEM VARCHAR2(20)
UPDATEUSER VARCHAR2(30)
UPDATEFUNCTION VARCHAR2(20)
UPDATEDATE DATE
STATUS VARCHAR2(15)
REVISIONNUMBER NUMBER(38)
FK_PERSONSBR_EUID
FK_ENTERPRISE_EUID
FK_SYSTEMSBR_EUID
SBYN_POTENTIALDUPLICATES
POTENTIALDUPLICATEID VARCHAR2(20) <pk>
WEIGHT VARCHAR2(20)
TYPE VARCHAR2(15)
DESCRIPTION VARCHAR2(120)
STATUS VARCHAR2(15)
HIGHMATCHFLAG VARCHAR2(15)
RESOLVEDUSER VARCHAR2(30)
RESOLVEDDATE DATE
RESOLVEDCOMMENT VARCHAR2(120)
EUID2 VARCHAR2(20)
TRANSACTIONNUMBER VARCHAR2(20)
EUID1 VARCHAR2(20)
SBYN_OVERWRITE
EUID VARCHAR2(20) <pk,fk>
PATH VARCHAR2(200) <pk>
TYPE VARCHAR2(20)
INTEGERDATA NUMBER(38)
BOOLEANDATA NUMBER(38)
STRINGDATA VARCHAR2(200)
BYTEDATA CHAR(2)
LONGDATA LONG
DATEDATA DATE
FLOATDATA NUMBER(38,4)
TIMESTAMPDATA DATE


SBYN_PERSONSBR
EUID VARCHAR2(20) <ak,fk>
PERSONID VARCHAR2(20) <pk>
PERSONCATCODE VARCHAR2(8)
LASTNAME VARCHAR2(40)
FIRSTNAME VARCHAR2(40)
MIDDLENAME VARCHAR2(30)
SUFFIX VARCHAR2(10)
TITLE VARCHAR2(8)
DOB DATE
DEATH VARCHAR2(1)
GENDER VARCHAR2(8)
MSTATUS VARCHAR2(8)
SSN VARCHAR2(16)
RACE VARCHAR2(8)
ETHNIC VARCHAR2(8)
RELIGION VARCHAR2(8)
LANGUAGE VARCHAR2(8)
SPOUSENAME VARCHAR2(100)
MOTHERNAME VARCHAR2(100)
MOTHERMN VARCHAR2(40)
FATHERNAME VARCHAR2(100)
MAIDEN VARCHAR2(40)
POBCITY VARCHAR2(30)
POBSTATE VARCHAR2(10)
POBCOUNTRY VARCHAR2(20)
VIPFLAG VARCHAR2(8)
VETSTATUS VARCHAR2(8)
FNAMEPHONETICCODE VARCHAR2(8)
LNAMEPHONETICCODE VARCHAR2(8)
MNAMEPHONETICCODE VARCHAR2(8)
MOTHERMNPHONETICCODE VARCHAR2(8)
MAIDENPHONETICCODE VARCHAR2(8)
SPOUSEPHONETICCODE VARCHAR2(8)
MOTHERPHONETICCODE VARCHAR2(8)
FATHERPHONETICCODE VARCHAR2(8)
DRIVERSLICENSE VARCHAR2(20)
DRIVERSLICENSEST VARCHAR2(10)
DOD DATE
DEATHCERTIFICATE VARCHAR2(10)
NATIONALITY VARCHAR2(8)
CITIZENSHIP VARCHAR2(8)
PENSIONNO VARCHAR2(15)
PENSIONEXPDATE DATE
REPATRIATIONNO VARCHAR2(16)
DISTRICTOFRESIDENCE VARCHAR2(8)
LGACODE VARCHAR2(4)
MILITARYBRANCH VARCHAR2(4)
MILITARYRANK VARCHAR2(4)
MILITARYSTATUS VARCHAR2(4)
DUMMYDATE DATE
CLASS1 VARCHAR2(20)
CLASS2 VARCHAR2(20)
CLASS3 VARCHAR2(20)
CLASS4 VARCHAR2(20)
CLASS5 VARCHAR2(20)
STRING1 VARCHAR2(40)
STRING2 VARCHAR2(40)
STRING3 VARCHAR2(40)
STRING4 VARCHAR2(40)
STRING5 VARCHAR2(40)
STRING6 VARCHAR2(40)
STRING7 VARCHAR2(100)
STRING8 VARCHAR2(100)
STRING9 VARCHAR2(100)
STRING10 VARCHAR2(255)
DATE1 DATE
DATE2 DATE
DATE3 DATE
DATE4 DATE
DATE5 DATE
STDFIRSTNAME VARCHAR2(40)
STDLASTNAME VARCHAR2(40)
STDMIDDLENAME VARCHAR2(30)
To SBYN_SYSTEMSBR by FK_PERSONSBR_EUID
FK_PHONESBR_PERSONID
SBYN_PHONESBR
PERSONID VARCHAR2(20) <ak,fk>
PHONEID VARCHAR2(20) <pk>
PHONETYPE VARCHAR2(8) <ak>
PHONE VARCHAR2(20)
PHONEEXT VARCHAR2(6)
FK_COMMENTSBR_PERSONID
SBYN_COMMENTSBR
PERSONID VARCHAR2(20) <ak,fk>
COMMENTID VARCHAR2(20) <pk>
COMMENTCODE VARCHAR2(8) <ak>
ENTERDATE DATE
COMMENTTEXT VARCHAR2(1000)
FK_AUXIDSBR_PERSONID
SBYN_AUXIDSBR
PERSONID VARCHAR2(20) <ak,fk>
AUXIDID VARCHAR2(20) <pk>
AUXIDDEF VARCHAR2(10) <ak>
ID VARCHAR2(40) <ak>
FK_ALIASSBR_PERSONID
SBYN_ALIASSBR
PERSONID VARCHAR2(20) <ak,fk>
ALIASID VARCHAR2(20) <pk>
LASTNAME VARCHAR2(40) <ak>
FIRSTNAME VARCHAR2(40) <ak>
MIDDLENAME VARCHAR2(30) <ak>
LNAMEPHONETICCODE VARCHAR2(8)
FNAMEPHONETICCODE VARCHAR2(8)
MNAMEPHONETICCODE VARCHAR2(8)
STDFIRSTNAME VARCHAR2(40)
STDLASTNAME VARCHAR2(40)
STDMIDDLENAME VARCHAR2(30)
FK_ADDRESSSBR_PERSONID
SBYN_ADDRESSSBR
PERSONID VARCHAR2(20) <ak,fk>
ADDRESSID VARCHAR2(20) <pk>
ADDRESSTYPE VARCHAR2(8) <ak>
ADDRESSLINE1 VARCHAR2(40)
ADDRESSLINE2 VARCHAR2(40)
ADDRESSLINE3 VARCHAR2(40)
ADDRESSLINE4 VARCHAR2(40)
CITY VARCHAR2(30)
STATECODE VARCHAR2(10)
POSTALCODE VARCHAR2(8)
POSTALCODEEXT VARCHAR2(4)
COUNTY VARCHAR2(20)
COUNTRYCODE VARCHAR2(20)
HOUSENUMBER VARCHAR2(10)
STREETDIR VARCHAR2(5)
STREETNAME VARCHAR2(40)
STREETNAMEPHONETICCODE VARCHAR2(8)
STREETTYPE VARCHAR2(5)


SBYN_TRANSACTION
TRANSACTIONNUMBER VARCHAR2(20) <pk,ak>
LID1 VARCHAR2(25)
LID2 VARCHAR2(25)
EUID1 VARCHAR2(20)
EUID2 VARCHAR2(20) <ak>
FUNCTION VARCHAR2(20)
SYSTEMUSER VARCHAR2(30)
TIMESTAMP DATE
DELTA LONG RAW
SYSTEMCODE VARCHAR2(20)
LID VARCHAR2(25)
EUID VARCHAR2(20) <ak>
FK_SBYN_MERGE
SBYN_MERGE
MERGE_ID VARCHAR2(20) <pk>
KEPT_EUID VARCHAR2(20) <fk>
MERGED_EUID VARCHAR2(20) <fk>
MERGE_TRANSACTIONNUM VARCHAR2(20) <fk>
UNMERGE_TRANSACTIONNUM VARCHAR2(20)
FK_AM_TRANSACTIONNUMBER
SBYN_ASSUMEDMATCH
ASSUMEDMATCHID VARCHAR2(20)
EUID VARCHAR2(20)
SYSTEMCODE VARCHAR2(20)
LID VARCHAR2(25)
WEIGHT VARCHAR2(20)
TRANSACTIONNUMBER VARCHAR2(20) <fk>
SBYN_AUDIT
AUDIT_ID VARCHAR2(20) <pk>
PRIMARY_OBJECT_TYPE VARCHAR2(20)
EUID VARCHAR2(15)
EUID_AUX VARCHAR2(15)
FUNCTION VARCHAR2(32)
DETAIL VARCHAR2(120)
CREATE_DATE DATE
CREATE_BY VARCHAR2(20)
SBYN_USER_CODE
CODE_LIST VARCHAR2(20) <pk>
CODE VARCHAR2(20) <pk>
DESCR VARCHAR2(50)
FORMAT VARCHAR2(60)
INPUT_MASK VARCHAR2(60)
VALUE_MASK VARCHAR2(60)
SBYN_COMMON_HEADER
COMMON_HEADER_ID NUMBER(10) <pk>
APPL_ID NUMBER(10)
CODE VARCHAR2(8)
DESCR VARCHAR2(50)
READ_ONLY CHAR
MAX_INPUT_LEN NUMBER(10)
TYP_TABLE_CODE VARCHAR2(3)
CREATE_DATE DATE
CREATE_USERID VARCHAR2(20)
FK_COMM_DET_COMM_HEAD
SBYN_COMMON_DETAIL
COMMON_DETAIL_ID NUMBER(10) <pk>
COMMON_HEADER_ID NUMBER(10) <fk>
CODE VARCHAR2(20)
DESCR VARCHAR2(50)
READ_ONLY CHAR
CREATE_DATE DATE
CREATE_USERID VARCHAR2(20)

Chapter 3

Installation

Before running the database migration, you must upload eIndex to the Repository and install it to the Enterprise Designer. This chapter provides instructions on installing the upgrade files once this has been done.

3.1 System Requirements

eIndex 5.0.4 supports different operating system, database, and integration software platforms than e*Index 4.5.3. If you need to upgrade any of the following components, be sure to complete the upgrades before beginning the upgrade. The following platforms are supported by eIndex 5.0.4.

Operating Systems

eIndex 5.0.4 supports the following operating systems.

- Windows Server 2003, Windows XP SP1a, and Windows 2000 SP3
- HP Tru64 V5.1A with patch 5
- HP-UX 11.0 and 11i with required patches and parameter changes
- IBM AIX 5.1 and 5.2 with required maintenance level patches
- Red Hat Enterprise Linux AS 2.1, Red Hat Linux 8 (Intel Version)
- Sun Solaris 8 and 9 with required patches

Important: *Ascential INTEGRITY 3.10 does not support LINUX. This platform is only available for implementations using the SeeBeyond Match Engine.*

Database Platforms

eIndex 5.0.4 supports Oracle 8.1.7 or 9*i*, but does not support Microsoft SQL Server or Sybase at this time. Upgrading to 5.0.4 requires Oracle 9*i* due to a change in data types that is only supported by Oracle 9*i* tools.

eGate Integrator

eIndex 5.0.4 can be installed with the eGate Integrator version 5.0.1 or later. The Oracle eWay is no longer required for eIndex, but can be used for database connectivity.

e*Index Global Identifier

Your current e*Index environment must be at version 4.5.3 in order to perform the migration.

Java Requirements for the Database Migration and Active Integration

For working with any of the Java classes, make sure you have access to the JVM packaged with the SeeBeyond Integration Server. You must have access to Java 1.4.0 or higher.

3.2 Installing the Upgrade Files

In order to work with the eIndex upgrade files, you must install eIndex. This provides the database scripts to create the upgraded database, and provides sample Projects to help you upgrade the 4.5.3 Schemas. Perform the following steps to download the upgrade files.

- **Step 1: Install eIndex 5.0.4** on page 20
- **Step 2: Download the Upgrade Files** on page 20
- **Step 3: Set the Environment Variables** on page 21

Step 1: Install eIndex 5.0.4

You need to install eIndex 5.0.4 before performing the database upgrade. Follow the instructions provided in chapter 3 of the *eIndex Global Identifier Composite Application User's Guide* to install eIndex. This chapter also provides information for installing documentation, Javadocs, and the INTEGRITY add-on. Make sure to install the INTEGRITY add-on if you will continue to use the INTEGRITY matching algorithm from Ascential.

Step 2: Download the Upgrade Files

To download the upgrade files, make sure you have completed the previous step by installing eIndex and the INTEGRITY add-on, if needed.

To download the upgrade files

1. If you are not currently logged into the Enterprise Manager, log on now.

   The **Upload System Component Manifest** window appears with the **HOME** tab active.

2. On the Enterprise Manager, click the **DOWNLOADS** tab.

The **products available to download from <*Repository_name*>** window appears.

3. Click **eIndex Upgrade**. You can either open the file and extract the upgrade files, or you can save the file to disk and extract the files at a later time.

Note: *Make sure to note the location to which you extract the files. You will need to set this as an environment variable and you will run the Java classes for the migration from this location.*

Step 3: Set the 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. If the migration server is running on a Windows operating system, make sure to specify the drive designation for each path.

To set environment variables

1. Complete **"Step 2: Download the Upgrade Files"**.
2. On the machine where the home directory resides, add the path to the Java home directory 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
3. Add the following information to the **CLASSPATH** variable. These files are all located in the directory to which you extracted the upgrade files in the **\lib** folder.
   - The absolute path and filename of the **classes12.jar** file in your Oracle environment.
   - The absolute path and filenames of the migration package **.jar** files. The **.jar** files are named **log4j.jar**, **stc_eindex_client.jar**, **stc_eindex_upgrade.jar**, and **stc_eindex_util.jar**.
   - The path to the properties files used by the migration package. These files are located in the **\<home_dir>\config** directory.
4. 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).

Chapter 4

Updating the eIndex Project

Before beginning the database upgrade in the production environment, you must configure the application using the files in the eIndex Project. This chapter provides general information about the types of updates to perform. Your implementation may require additional updates to the eIndex Project.

4.1 General Updates

This section describes general changes you need to make to the eIndex Project before performing the database migration. These changes will help ensure the eIndex environment is properly configured and ready to go live as soon as the database migration is complete. This list might not be complete for your implementation. See the *eIndex Global Identifier Composite Application User's Guide* to learn about the possible customizations you can make to the eIndex Project.

4.1.1 eIndex Configuration

As discussed in Chapter 2, the configuration for eIndex is stored in a set of XML files in the eIndex Project. Modify these files to customize the configuration of the master index. This includes setting weight thresholds, date formats, field attributes, queries, matching logic, and so on. For more information about what can be customized, see the *eIndex Global Identifier Composite Application Configuration Guide*. For upgrades, do not modify the object structure.

4.1.2 Defining a Data Connection

You can connect to the eIndex 5.0.4 database using a JDBC connection pool or using an Oracle eWay. Either way, you must define the properties for the database in order for eIndex to be able to connect to the database. Both methods are described in the *eIndex Global Identifier Composite Application User's Guide*.

4.1.3 Customizing Connectivity

The eIndex Project includes a predefined Connectivity Map that you can customize for use in your production Project. eIndex also includes a client Project designed to transfer information between the master index and external systems. This Project includes a

Collaboration with defined processing rules that are similar to those defined in the **uidb.dsc** file in version 4.5.3. The primary difference between the two Collaborations is the change from Monk in version 4.5.3 to Java in version 5.0. You can customize this Collaboration to match the processing rules you defined for version 4.5.3.

The client Project also includes a default input OTD based on the default object structure. For more information about eIndex Connectivity Maps, see the *eIndex Global Identifier Composite Application User's Guide*. For more information about Connectivity Maps in general, see the *eGate Integrator User's Guide*.

4.1.4 Defining the Environment and Deployment Profile

Before working with eIndex, you must define the physical environment in the Enterprise Designer. Once this is defined, you can create and activate the deployment profile for the eIndex Project. In order to perform the final step, you should have a logical host installed. eIndex Environments and Deployment Profiles are described in the *eIndex Global Identifier Composite Application User's Guide*.

4.1.5 Setting up Security

Any security information from version 4.5.3 must be manually defined in the eIndex Environment. Like 4.5.3, you can add users and assign them specific functional permissions; however, the permission categories and names have changed between versions. Refer to the *eIndex Global Identifier Composite Application User's Guide* for more information about the new access permission names and setting up security.

4.2 Updating the Unmerge Policy

After you have performed the upgrade and have the eIndex Project installed in Enterprise Designer, you can modify the unmerge policy to enable unmerging records that were merged in previous versions. The upgrade package includes an updated unmerge policy that you can use to replace the default unmerge policy. If you want to retain the ability to unmerge records in version 5.0.4 that were merged in version 4.5.3, you must update the default enterprise unmerge policy.

To update the unmerge policy

1. In the Enterprise Designer, expand the eIndex server Project (named "eIndex" by default), expand the eIndex application, and then expand **Custom Plug-ins**.
2. Right-click **EnterpriseUnmergePolicy**, and then select **Check Out**.
3. Double-click **EnterpriseUnmergePolicy** to open the file in the editor.
4. Delete all text in the file.
5. On your desktop, navigate to the directory to which you extracted the upgrade package, and then to the **\policy** folder.
6. Open the file **EnterpriseUnmergePolicy.java** in any text or XML editor.

7 Copy the entire text of the file, and paste the text into the **EnterpriseUnmergePolicy** custom plug-in in the Enterprise Designer.

8 Save the changes to the **EnterpriseUnmergePolicy** custom plug-in and check the file back into the Repository.

9 Regenerate the eIndex application.

10 If you have already deployed the application, you must reactivate the deployment profile and apply the changes to the Logical Host.

Chapter 5

Migrating the Database

This chapter presents the background information and the step-by-step instructions you need to migrate your eIndex database from version 4.5.3 to version 5.0.4.

5.1 Learning About the Properties Files

The properties files define the runtime characteristics for the Java API used for the migration process. Several default properties files are installed with the migration package. You need to modify these files for your processing environment (this is described later in this chapter). The following properties files are used during the data migration.

- **migration.properties**
- **source.properties**
- **target.properties**
- **log4j.properties**

migration.properties

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** file contains these variables.

- **maxErrors**
  The number of errors allowed during a migration run before the process will abort.
- **reprocess**
  An indicator of whether the current run is in reprocess mode (that is, error records from a previous run are being reprocessed). When processing new records, enter **false**; when reprocessing error records, enter **true**.

source.properties

The **source.properties** file defines properties about the 4.5.3 source database, which is the copy you will make of your production database. The properties define information about the database, 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 connections, tracing levels, and so on. The **source.properties** file defines the following variables.

- **databaseMaximumConnections**
  The maximum number of database connections that may be used at one time.
- **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**.
- **databaseUserId**
  The user name with which to log on to the e*Index 4.5.3 source database. The user ID specified must have the appropriate access permissions assigned.
- **databasePassword**
  The password associated with the specified **databaseUserId** for the 4.5.3 source database.
- **databasePortNumber**
  The port number where a server is listening for requests from the Java program for the 4.5.3 source database.
- **databaseServerName**
  The name of the database server on which the 4.5.3 source database is running. This value can be the name or IP address of the server.
- **databaseVendor**
  The database platform being used for the 4.5.3 source database. Currently only Oracle is supported for the migration package.
- **databaseName**
  The Oracle SID name of the 4.5.3 source database.

target.properties

The **target.properties** file defines properties about the 5.0.4 target database, which is the database that will become the production database in the eIndex 5.0.4 environment. The properties define information about the database, 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 connections, tracing levels, and so on. The **target.properties** file defines the following variables.

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

- **databaseUserId**
  The user name with which to log on to the eIndex 5.0.4 target database. The user ID specified must have the appropriate access permissions assigned.
- **databasePassword**
  The password associated with the specified **databaseUserId** for the 5.0.4 target database.
- **databasePortNumber**
  The port number where a server is listening for requests from the Java program for the 5.0.4 target database.
- **databaseServerName**
  The name of the database server on which the 5.0.4 target database is running. This value can be the name or IP address of the server.
- **databaseVendor**
  The database platform being used for the 5.0.4 target database. Currently only Oracle is supported for the migration package.
- **databaseName**
  The Oracle SID name of the 5.0.4 target database.

log4j.properties

The **log4j.properties** file defines properties for the log files generated by the upgrade process. The only **log4j.properties** file variables you might need to modify are the following:

- **log4j.appender.FILE.File**
  The default location to which the log file is written. Change the default value to a location on your computer.
- **log4j.appender.FILE.MaxFileSize**
  The maximum size to which the log file can grow. The default is **10MB**.
- **log4j.appender.FILE.MaxBackupIndex**
  The maximum number of backup file. The default is **10**.
- **log4j.appender.FILE.layout**
  The class that defines the layout format of the log files. The default is **org.apache.log4j.PatternLayout**.
- **log4j.appender.FILE.layout.ConversionPattern**
  Defines attributes of the log file, such as the date format (%d), indentation of the debug level (%p), category (%c), and the nested diagnostic context (%x). The default is **%d(ISO8601) %-5p [%c] [%x]**.
- **log4j.rootCategory**
  The level of debugging information to record and the location to which it will be printed. Possible levels are as follows:
  - **DEBUG** - Indicates information comments that might be useful in debugging the upgrade process.

- **WARN** - Indicates events that do not qualify as errors, but that may be harmful to the upgrade process.
- **INFO** - Indicates informational comments that record the progress of the upgrade process.
- **ERROR** - Indicates errors that are not severe enough to cause the upgrade process to abort.
- **FATAL** - Indicates severe errors that cause the upgrade process to abort.

The output value is **FILE**. This should not be changed.

5.2 Setting up the Databases

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 eIndex 5.0.4 database and then switch the new database over to production. Complete the following steps to create the migration databases.

- **Step 1: Create and Configure the Source Database** on page 28
- **Step 2: Create the eIndex 5.0.4 Target Database** on page 28
- **Step 3: Begin Queuing 4.5.3 and 5.0.4 Daily Transactions** on page 29
- **Step 4: Back up the Production Database** on page 29
- **Step 5: Export and Import the Production Data** on page 29
- **Step 6: Restart the 4.5.3 eWays** on page 29

Step 1: Create and Configure the Source Database

Following the instructions in the installation guide for e*Index 4.5.3, create a new Oracle database that is identical to your current production database. If possible, use the same installation scripts that you used for your 4.5.3 database. This will be the source database, and will be used instead of the current production database for the migration.

After you create the database, run the following scripts. These files are located in the directory to which you extracted the upgrade files in the **\db\oracle** folder.

- create_error_table.sql
- create_views.sql
- dm_job_rows.sql
- dm_parent_job_rows.sql

Step 2: Create the eIndex 5.0.4 Target Database

Creating the 5.0.4 database involves creating a standard Oracle database and then running the eIndex scripts against the database. Perform the steps described in chapter

7, "Creating the Database" of the *eIndex Global Identifier Composite Application User's Guide* for instructions on creating the database. This chapter includes instructions for defining systems and common table data, but you do not need to perform these steps. This information is automatically transferred by the upgrade package (you will need to update the migrated data). This means you do not need to customize or run the **Systems** or **Code List** scripts. To successfully complete the upgrade, the target database must be an Oracle 9*i* database.

The eIndex 5.0.4 target database will become your production database once all data is migrated. You can install this database on the same server that hosts the 4.5.3 source database if the server's processing capabilities are powerful enough.

Step 3: Begin Queuing 4.5.3 and 5.0.4 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 eIndex 5.0.4 Project should be tested and ready for production, and you need to queue records from external systems into both the 4.5.3 schema and 5.0.4 Project. This will allow you to continue to use the 4.5.3 database while at the same time queuing up the live data for the 5.0.4 database.

Step 4: 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 your Oracle8*i* documentation.

Step 5: Export and Import the Production Data

The final step in setting up the databases is to export the existing data in the production database and then import that data into the source database. You can use the standard Oracle import and export functions to perform these tasks.

Step 6: Restart the 4.5.3 eWays

Once the databases are set up, you can restart the 4.5.3 eWays to process data through the production database.

5.3 Migrating the Database

To migrate the data from your source 4.5.3 database to the target eIndex 5.0.4 database, you need to complete the following steps.

- **Step 1: Modify the Properties Files** on page 30
- **Step 2: Perform the Initial Data Transfer** on page 30
- **Step 3: Migrate History Data** on page 31

- **Step 4: Update the Sequence Table** on page 32
- **Step 5: Review the Migration Process** on page 32
- **Step 6: Catch up from eGate Queue** on page 32

Step 1: Modify the 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 four database properties files to configure.

To modify the properties files

1. Make sure you have completed all of the procedures in the preceding section **"Setting up the Databases"**.
2. Navigate to the migration package home directory, and then to **\config**.
3. Open the file **target.properties** in any text editor and modify the variables in the file for the 5.0.4 target database. For a complete list of variables and their definitions, see **"target.properties" on page 26**.
4. Save and close **target.properties.**
5. Open the file **source.properties** in any text editor and modify the variables in the file for the 4.5.3 source database. For a complete list of variables and their definitions, see **"source.properties" on page 25**.
6. Save and close **source.properties**.
7. Open the file **migration.properties** in any text editor and modify the variables in the file. For a complete list of variables and their definitions, see **"migration.properties" on page 25**.

Note: *For the initial run, make sure the* ***reprocess*** *property is set to* ***false****.*

8. Save and close **migration.properties**.
9. Open the file **log4j.properties** in any text editor and modify the variables in the file. For a list of variables you might want to change and their definitions, see **"log4j.properties" on page 27**.
10. Save and close **log4j.properties**.
11. Continue to **"Step 2: Perform the Initial Data Transfer"**.

Step 2: Perform the Initial Data Transfer

The migration is performed in steps, current person data being processed first. This step also migrates common table data, system information, potential duplicates, assumed match information, and so on. This step can take quite a while, depending on the size of the database.

To perform the initial data transfer

1. Complete **"Step 1: Modify the Properties Files"**.

2 Run the following Java command:

```
java com.stc.eindex.upgrade.TableMigration
```

3 Review the log file created for this step. The location of the log file is specified in the **log4j.properties** file.

4 Review the exception table *dm_error* to see if there were any errors in processing the data. If the table lists errors, do the following:

A Fix the bad data in the data tables of the source database.

B Change the value of the **reprocess** property in the **migration.properties** file to **true**.

C Set the value of the reprocess column in the dm_error table to "Y" for each record you want to reprocess.

D Rerun the table migration as described in step 2 above.

E Repeat the above steps until all records are processed.

5 Continue to **"Step 3: Migrate History Data"**.

Step 3: Migrate History 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 can take several days depending on the size of your database.

To migrate history data

1 Complete **"Step 2: Perform the Initial Data Transfer"**.

2 Make sure the value of the **reprocess** property in the **migration.properties** file to **false**.

3 Run the following Java command:

```
java com.stc.eindex.upgrade.DeltaMigration
```

4 Review the log file created for this step. The log file is created in the folder you specified in the **log4j.properties** file.

5 Review the exception table *dm_error* to see if there were any errors in processing the data. If the table lists errors, do the following:

A Fix the bad data in the history tables of the source database.

B Change the value of the **reprocess** property in the **migration.properties** file to **true**.

C Set the value of the reprocess column in the dm_error table to "Y" for each record you want to reprocess.

D Rerun the delta migration as described in step 3 above.

E Repeat the above steps until all records are processed.

6 Continue to **"Step 4: Update the Sequence Table"**.

Step 4: Update the Sequence Table

The final upgrade function updates the *sbyn_seq_table* database table.

To update the sequence table

1. Complete **"Step 3: Migrate History Data"**.
2. Run the following Java command:

```
java com.stc.eindex.upgrade.SequenceTableUpdater
```

3. Continue to **"Step 5: Review the Migration Process"**.

Step 5: Review the Migration Process

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

- Verify that the *ui_person* table in the source databases contains the same number of active records as the *sbyn_personsbr* table in the target database (deactivated and merge records are stored in *ui_person*, but not in *sbyn_personsbr*). You may need to take into account the number of records that could not be migrated. These are listed in a table named dm_bad_uid.
- Spot check individual person records by comparing the records using either the eIndex 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 EUIDs to verify the information.
- Spot check individual records in *ui_person* for accuracy.

Step 6: Catch up from eGate Queue

Once you have reviewed the migration process and made any necessary adjustments to the database, you can bring the 5.0.4 Project live and start processing the queued data and live feeds into the database. Monitor this process closely in the beginning to ensure that data is being processed correctly.

Index

A

active integration **11**
administrative data **10**

B

blocking query **11**

C

Candidate Select
  configuration file **11**
CLASSPATH **21**
client workstation **10**
Collaboration **23**
common table data **10–11**
configurable query **10–11**
configuration file **10**
configuration files **22**
Connectivity Map **22**
control keys **10–11**
create_error_table.sql **28**
create_views.sql **28**
customization to the database **13–14**

D

database
  4.5.3 to 5.0.4 mapping **12**
  defining connectivity **22**
  obsolete tables **13**
  supported platforms **19**
  upgrade methodology **12**
  upgrade overview **12–14**
  upgrade phases **12**
  upgrading customizations **13–14**
database diagram **14–18**
databaseDriverType **26**
databaseMaximumConnections **26**
databaseName **26, 27**
databaseNetworkProtocol **26**
databasePassword **26, 27**
databasePortNumber **26, 27**
databaseServerName **26, 27**
databaseUserId **26, 27**
databaseVendor **26, 27**
date formats **22**
DEBUG **27**
deployment profile **23**
dm_job_rows.sql **28**
dm_parent_job_rows.sql **28**
document conventions **7**
duplicate check **11**

E

e*Index Administrator **10**
e*Index Security **10**
EDM
  see Enterprise Data Manager
eGate Integrator **19**
eGate queues **14, 29, 32**
Enterprise Data Manager **10**
Enterprise Designer **23**
Enterprise Manager **20**
environment **23**
environment variables **21**
ERROR **28**
error messages **31**
event notification **10**
eWays **29**
extensive searching **11**

F

FATAL **28**
field attributes **22**

H

history data **13**

I

INFO **28**
Integration Server **10**
INTEGRITY **10–11**

J

Java **9, 11, 21, 23**
JMS Topic
  upgrading to **10**

L

LD_LIBRARY_PATH **21**

LIBPATH **21**
log file **31**
log4j.appender.FILE.File **27**
log4j.appender.FILE.layout **27**
log4j.appender.FILE.layout.ConversionPattern **27**
log4j.appender.FILE.MaxBackupIndex **27**
log4j.appender.FILE.MaxFileSize **27**
log4j.properties **27–28**, **30**
log4j.rootCategory **27**

M

Match Field
  configuration file **11**
maxErrors **25**
migration.properties **25**, **30**
Monk **9**, **23**

O

obsolete tables **13**
one exact match **11**
online documents **8**
operating systems **19**
outbound queue
  upgrading **10**

P

PATH **21**
production database **12**, **28**
  backing up **29**
Project
  configuration files **22**
  upgrading to **9**
Project updates **22**
properties files
  log4j.properties **27–28**
  migration.properties **25**
  modifying **30**
  source.properties **25–26**
  target.properties **26–27**

Q

queries **22**
Query Manager **11**
queueing **14**, **29**, **32**
queueing messages **12**

R

reports **11**
reprocess **25**, **31**
reprocessing records **31**
requirements **19–20**
rule set files **10–11**

S

sbyn_transaction **13**
schema
  upgrade overview **9**
security **10**, **23**
  upgrading **10**
SeeBeyond Web site **8**
SHLIB_PATH **21**
source database **12**, **28**
source.properties **25–26**, **30**
system requirements **19–20**

T

target database **12**, **28–29**
target.properties **26–27**, **30**
thresholds **11**, **22**
Topic, JMS
  upgrading to **10**

U

uidb.dsc **23**
unmerge policy
  updating **23–24**
upgrade requirements **19–20**
upgrading
  the schema **9**

W

WARN **28**

X

XML **22**