*e*Index Global Identifier Product Suite*

e*Index™ Global Identifier Release Bulletin

Version 4.5.2

SEEBEYOND

404 East Huntington Drive, Monrovia, CA 91066 (626) 471-6000 http://www.seebeyond.com

Table of Contents

Chapter 1

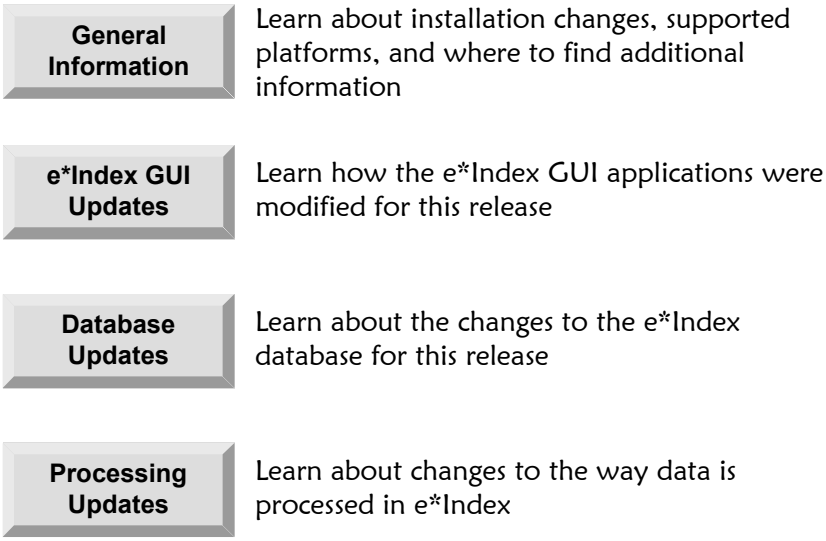
Release Information for Release 4.5.2

About this Bulletin

Overview

This release bulletin for e*Index Global Identifier 4.5.2 describes the differences between e*Index 4.5.1 and 4.5.2. For detailed information on any of the functionality covered in this document, please refer to the supporting documentation or online Help.

The following diagram illustrates the contents of each major topic in this chapter.



General Information

Overview

This section of the chapter provides general information about supported operating system and database platforms, as well as the required SeeBeyond software for e*Index 4.5.2.

Where to Find More Information

When you upgrade to e*Index 4.5.2, you automatically install updated online Help systems; you can find information about GUI features in each GUI's Help system.

Updated user's guides can be found in the electronic library for e*Index. These guides can be installed from the e*Index CD-ROM. User's guides are provided in PDF format, and require Adobe® Acrobat® Reader (available free over the Internet at http://www.adobe.com).

Available Platforms

e*Index Global Identifier 4.5.2 requires the following software from SeeBeyond and other vendors, and can operate on the operating systems listed below.

SeeBeyond Products

e*Index now requires the following versions of e*Gate and the Database e*Ways:

- e*Gate Integrator 4.5.0 or later
- Oracle e*Way, Sybase e*Way, or ODBC e*Way, depending on the database platform (all version 4.5.0 or later)
- HL7 Templates for e*Gate 4.5.0 or later (only if you send HL7 messages)
- e-Mail e*Way 4.5.0 or later (only if you use the Event Notification function)

Database Software

There are no changes for database software support for this release, however the recommended version of Oracle is 8.1.7.2.1. The supported database platforms are as follows:

- **Oracle**
  Oracle software must be installed as follows in order to install the e*Index database on an Oracle database platform.
  - Oracle 8.1.7 Server on the database server
  - Oracle 8.1.7 Client on the client workstations and e*Gate host or participating host
- **Sybase**
  Sybase software must be installed in one of the following combinations in order to install the e*Index database on a Sybase database platform.
  - Sybase 12.0 Server on the database server
  - Sybase 12.0 Client on the client workstations and e*Gate host or participating hosts
- **Microsoft SQL Server**
  Microsoft SQL Server software must be installed as follows in order to install the e*Index database on a Microsoft SQL Server database platform.
  - Microsoft SQL Server 7.0 on the database server
  - Microsoft SQL Server 7.0 (client components only) on the client workstations and e*Gate host or participating hosts

Operating Systems

e*Index 4.5.2 supports two additional operating system versions: HPUX 11i and AIX 5.1. Currently, the e*Gate schema for e*Index is supported on the following platforms:

- Windows NT 4.0 SP4 or higher, Windows 2000 SP2
- Solaris 2.6 or later
- HPUX 11 or 11i
- AIX 4.3.3 or 5.1
- TRU64 V4.0F and V5.0A

The e*Index client workstation is supported on the following platform:

- Windows 2000 SP2
- Windows NT 4.0 SP4 or later
- Windows 95
- Windows 98

New Upgrade Package

Overview

The most significant changes to e*Index upgrades is the addition of the upgrade package from version 4.1.1 to version 4.5.2. This section provides background information about the upgrade package.

About the Upgrade Process

The upgrade process from e*Index 4.1.1 to 4.5.2 is different from the standard upgrade process between versions. The Schema and GUI upgrades are standard, but the database upgrade is a complex task due to the differences in database structures between the two versions. Rather than upgrading an existing database, the upgrade involves migrating data from a copy of the 4.1.1 production database to a new 4.5.2 database. During the migration, all of the person data is migrated from the 4.1.1 *ui_history* table into the person data tables of the 4.5.2 database using timestamp and UID information to link associated information. The database upgrade is performed using a combination of SQL and Java objects.

For More Information

For detailed information about this upgrade, see the *e*Index 4.1.1 to 4.5.2 Upgrade Guide*. For detailed information about the differences between versions 4.1.1 and 4.5.2, see the *e*Index 4.1.1 to 4.5.2 Release Bulletin*.

e*Index Global Identifier GUI Enhancements

Overview

Minor changes were made to the e*Index Global Identifier and Administrator GUIs for this release. The documentation was also modified to further clarify information about certain functions. This section describes those changes.

Audit Trail

The Audit Trail function has been enhanced to allow you to view complete histories for both merged and deactivated member profiles. Previously, you could only view the current image for the unkept profile after a merge. Audit trails for deactivated profiles were not accessible.

Non-unique IDs

In previous versions with a Sybase database implementation, non-unique IDs could not be updated due to a column type conflict. For this version, the ID update function was modified to allow non-unique ID updates.

Case-sensitive Entry and Search for Local IDs

Previously, alphabetic characters in local IDs could only be entered in uppercase from the GUI. In addition, searches for local IDs could only be performed in uppercase (even if a lowercase letter was entered, that letter would be converted to uppercase for the search). If a local ID was entered from the backend with a lowercase letter, users could not search for that local ID from the e*Index GUI. This release provides the ability to enter and search for local IDs using mixed cases. The types of characters (numeric, uppercase alphabetic, lowercase alphabetic, and so on) that can be entered for a local ID is based on the format definition for that local ID in the System Maintenance function of e*Index Administrator.

Checksum Functionality

In previous versions, the checksum functionality was not working as expected. This has been addressed in this release. The following paragraphs describe how the checksum functionality, controlled by the CKSUMLEN control key, works.

The *CKSUMLEN* control key allows you to specify the length of the checksum value. Checksum values help validate UIDs in order to ensure accurate identification of member profiles throughout the system. The

checksum process attaches a 2-digit number, generated through an algorithm, to the end of a new UID in order to validate that UID when transmitting messages. When a host system receives this number, it strips off the last two digits to find the UID, and then recalculates the checksum using the same algorithm process. If the checksums agree, the host system knows the UID number is correct.

The CHECKSUM control key works with the UIDLENGTH control key in the following manner. UIDLENGTH specifies the length of the sequential UID generated in the *ui_seq_no* table. However, when you specify a CHECKSUM length greater than 0, e*Index creates sequential UIDs based on the *ui_seq_no* table, and then appends the check sum value to the end of the UID to determine the final UID number. For example, if you set UIDLENGTH to 8 and CHECKSUM to 2, then the UIDs assigned by e*Index will be 10 characters long. If the next sequence number is **10908000**, then the UID assigned to the next record is the **10908000** plus the check sum (**1090800034**, for example). The next UID would be **10908001** plus the check sum (**1090800125**, for example). The first 8 digits are sequential, but the last 2 digits are arbitrary, based on the checksum algorithm.

Zip Code Maintenance

In previous versions, you could only add one zip code for each Zip Code Maintenance session in the e*Index Administrator. For this release, the Zip Code Maintenance function has been modified to allow multiple zip code additions per session.

Extensive Searching

The EXTNSVSRCH control key, which determines whether alphanumeric searches from the GUI include alias names as well as primary names, does not control phonetic searches or the searches performed when selecting the candidate selection pool for potential duplicate evaluation. To enable extensive searching for these two types of searches you need to modify the configurable query for both types. The *e*Index Administrator User's Guide* has been updated with complete instructions for enabling extensive searches, including the SQL code for the WHERE clause. For more information, see "Configuring Extensive Searching" in chapter 5 of the *e*Index Administrator User's Guide*.

MIXEDCASE Control Key

No changes were made to the MIXEDCASE control key for this release, but information about how the value of this control key affects the user IDs and passwords was updated in the *e*Index Administrator User's Guide*. The guide now states that this control key applies to all windows in all three e*Index

applications, including the Login and Change Password windows. Make sure that if this control key is set to **N** all users know to enter their passwords in uppercase characters when they log on or change their passwords.

e*Index Database Enhancements

Overview

This section describes the changes made to the e*Index database for release 4.5.2. For more detailed information about the e*Index database and to view a physical data model of the Oracle database, see Chapter 2 of the *e*Index Global Identifier Technical Reference* (Sybase and SQL Server data models are provided in chapters 4 and 5 of the *e*Index Global Identifier Upgrade Guide*).

Installation Scripts

For this release, two installation scripts, **defs.sql** and **initSID.ora**, were modified to enhance the sizing and distribution configuration of the default installation of the e*Index database.

- **defs.sql**
  The default value of the log file size (LFS) variable changed from 1M to 4M, and log file paths were modified from **&DB_HOME\&SID** to **&DB_HOME\&SID\database**.
- **initSID.ora**
  The following variables were modified for **initSID.ora**.
  - **db_block_buffers** was increased from 200 to 2000.
  - **shared_pool_size** was increased from 6500000 to 20000000.
  - **log_checkpoint_interval** was increased from 10000 to 1124000.
  - **log_buffer** was increased from 8192 to 1024000.
  - Two new variables were added: **sort_area_size** (with a value of 131072) and **open_cursors** (with a value of 100).

Reports

The monthly e*Index report files **monthly_c** and **monthly_p** were modified for this release. They previously contained a date argument that prevented the reports from running correctly. For this release, that date argument was removed and the reports run as expected. The excerpt below indicates the clause that was modified. The third line of this clause was removed.

```
ui_monthly_stat_proc(to_char(sysdate,'MONTH'),
        to_char(sysdate,'YYYY'));
        to_char(sysdate,'YYYYMM');
```

In addition, changes were made to the **rep_unmerge_t** and **rep_unmerge_y** report files. Previously these reports were printing duplicate entries of unmerge transactions. For the current release, only one instance of each

transaction appears on the reports. If you are upgrading your e*Index environment, you need to modify the four report files with the updates SQL statements.

County Field

In previous versions of e*Index, the county field in various tables in the e*Index database varied from 3 to 20 characters. The length of the field has been standardized to 20 characters in all applicable database tables.

Triggers

In previous versions with Sybase or Microsoft SQL Server implementations, the ui_seq_no trigger caused problems when using the checksum functionality (that is, the CKSUMLEN is set to a value other than **0** (zero)). This issue was addressed for this release by changing the seq_no type to numeric(15) for both Sybase and SQL Server.

e*Index Processing Modifications

Overview

This section of the chapter describes the changes made to the way data is processed in e*Index. The changes affect address-parsing for Australia addresses, the out queue, and the e*Index and initial load Schemas.

Vality INTEGRITY

In previous releases, the Australia address-parsing rule set interpreted PO Box as a street type rather than a street name. This causes "PO Box" to be truncated and the entire address record to be ignored. For this release, this issue was fixed by modifying the address-parsing options in Country Specific Option Maintenance in the e*Index Administrator.

Out Queue

In previous versions, the out queue (the *ui_msg_detail* table) stored duplicate sets of Events for the polling e*Way when an unmerge transaction occurred, and the polling e*Way extracted two identical Events for transmission to external systems. For this release, the second unmerge Event is no longer stored, and unmerge transactions result in one unmerge Event being stored in the out queue for the polling e*Way.

e*Index and Initial Load Schema

For this release, the **ui-process-person.monk** file for the e*Index Schema and the **ui-load-person.monk** file for the initial load Schema were modified to enhance performance. These files were modified by including the calls to ui-search-insert-duplicate and ui-search-close within "begin" blocks to ensure that the search cursor is being closed properly. SeeBeyond recommends updating your files with the new "begin" blocks.