*e*Index Global Identifier Product Suite*

e*Index™ Global Identifier Release Bulletin

Version 4.5.3

SEEBEYOND

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

Copyright

Version 453.200208

Table of Contents

Chapter 1: Release Information for e*Index 4.5.3 1-1

About this Bulletin 1-1

Overview 1-1

General Information 1-2

Overview 1-2

Where to Find More Information 1-2

Installation and Upgrade Changes 1-2

Available Platforms 1-2

SeeBeyond Products 1-3

Database Software 1-3

Operating Systems 1-3

Core Product Updates 1-5

Overview 1-5

Database Changes 1-5

Codeset File Changes 1-5

Active Integration Tables 1-5

Java APIs for e*Index Active Integration 1-7

Overview 1-7

Codeset Implementation 1-7

New Class Hierarchy 1-7

New Classes in the Person Package 1-8

New Interface in the Person Package 1-9

Alias Class 1-9

CodeDisplay Class 1-9

ControlKey Class 1-10

Demographics Class 1-10

New Methods 1-10

Obsolete Methods 1-11

Modified Methods 1-11

DemographicsResultsRO Class 1-12

New Methods 1-12

Obsolete Methods 1-13

Modified Methods 1-13

DemographicsRO Class 1-14

New Methods 1-14

Obsolete Methods 1-15

Modified Methods 1-15

EiBOFactory Class 1-16

EiConnection Class 1-16

EiServer Class 1-16

EiSystem Class 1-16

EnumCodeType Class 1-17

EnumDemographicField Class 1-17

EnumLocalIdStatus Class 1-17

LocalId Class 1-17

LocalIdBO Class 1-17

Person Class 1-17

PersonBO Class 1-18

SearchParameters Class 1-18

Security Class 1-19
Ssn Class 1-19
Transaction Class 1-19
Uid Class 1-19
Instantiated Classes 1-19
Exception Package 1-19
New Exception Classes 1-19
Obsolete Exception Classes 1-20
Log Files 1-21

Chapter 1

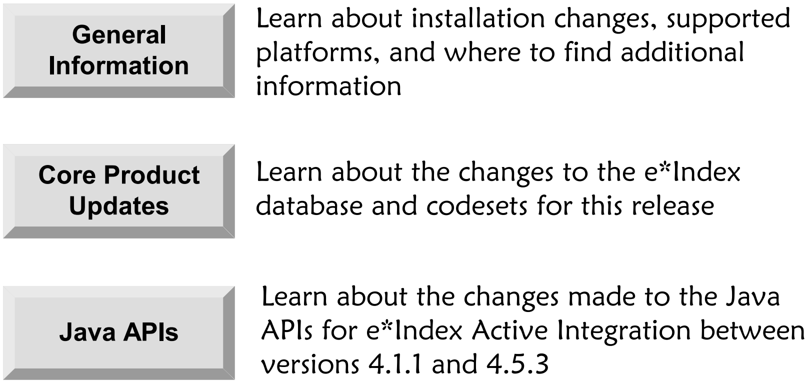
Release Information for e*Index 4.5.3

About this Bulletin

Overview

This release bulletin for e*Index Global Identifier 4.5.3 describes the differences between e*Index 4.5.3 and 4.5.2, and the differences between the Java API for e*Index Active Integration version 4.1.1 and version 4.5.3. For detailed information on any of the functionality covered in this document, please refer to the supporting documentation or online Help. For information about the changes between versions 4.1.1 and 4.5.2, see the *e*Index 4.1.1 to 4.5.2 Release Bulletin.*

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



General Information: Learn about installation changes, supported platforms, and where to find additional information

Core Product Updates: Learn about the changes to the e*Index database and codesets for this release

Java APIs: Learn about the changes made to the Java APIs for e*Index Active Integration between versions 4.1.1 and 4.5.3

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 4.5.3.

Where to Find More Information

This release of e*Index provides a set of Java APIs for e*Index Active Integration. Updated and new e*Index manuals can be found in the electronic library for e*Index. New guides include the *Java Programmer's Reference for e*Index Active Integration*; updated guides include the e**Index Global Identifier Installation Guide*, the *e*Index Global Identifier Upgrade Guide*, the *e*Index Administrator User's Guide*, the *e*Index Global Identifier Technical Reference*, and the *e*Index Global Identifier Release Bulletin*. 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).

If you are upgrading from e*Index 4.1.1 and are using the Java APIs for e*Index Active Integration, SeeBeyond recommends reviewing the *e*Index 4.1.1 to 4.5.2 Release Bulletin*. This guide describes the new functionality incorporated into e*Index to make the product more flexible and customizable. Understanding these changes will help you understand the changes made to the Java APIs for this release.

Installation and Upgrade Changes

The Java APIs for e*Index Active Integration are installed from an InstallShield separate from the core product InstallShield. The setup files for the Java APIs are included with the standard installation files in a folder named **/active**. For information about how to install or upgrade the Java APIs, see chapter 7 of the *e*Index Global Identifier Installation Guide* or the *e*Index Global Identifier Upgrade Guide*.

Available Platforms

e*Index Global Identifier 4.5.3 requires the following software from SeeBeyond and other vendors, and can be installed 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, the e*Gate host or participating host, and the server hosting the Java API
- **Sybase**
  Sybase software must be installed as follows 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

Important! *The Java APIs for e*Index Active Integration are only supported on an Oracle database. If you will be using the Java APIs, be sure to implement the e*Index database on an Oracle platform.*

Operating Systems

e*Index 4.5.3 supports two additional operating system versions: HPUX 11i and AIX 5.1. e*Index no longer supports Solaris 2.6 or HP-UX 10.0. Currently, the e*Gate Schema for e*Index is supported on the following platforms:

- Windows NT 4.0 SP6
- Windows 2000 SP2
- Solaris 2.7 and 2.8
- HP-UX 11 and 11i
- AIX 4.3.3 and 5.1 (note that Oracle client, which must be installed for the e*Index Schema, is only supported with the 32-bit version of AIX 5.1)
- Compaq Tru64 v4.0f or v5.1a

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

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

Requirements for the Java APIs for Active Integration

The following software must be installed on the computer on which the APIs will reside. Note that the Java APIs are only supported with an e*Index database implemented on the Oracle database platform.

- **Supported operating systems:**
  Windows NT 4.0 SP6, Windows 2000, HP Unix 11.0 or 11i, IBM AIX 4.3.3 or 5.1, Compaq Tru64 v4.0f or v5.1a, or Solaris 2.7 or 2.8
- **Required database software:**
  Oracle Client version 8.1.7
- **Required Java software:**
  Java™ 2 SDK, Standard Edition 1.1.8 or higher (for HP Unix 11.0, you must install version 1.2 or higher; for all other platforms, version 1.3.1 is recommended)
- **Required browser software:**
  A web browser compatible with Javadocs (to view the Javadocs for the Java APIs for e*Index Active Integration)

Core Product Updates

Overview

Minor changes were made to the e*Index database and codesets for the standard implementation of e*Index. For the Java APIs for e*Index Active Integration, there are several new tables you can add to the database for added customizations.

Database Changes

For this release, minor changes were made to the **ui_start_transaction** routine. To add this update to your e*Index database, follow the instructions for updating the appropriate database platform in the *e*Index Global Identifier Upgrade Guide*.

Codeset File Changes

To ensure that the e*Ways and client workstations are using a standard set of codesets for Vality data mapping, the codesets are now downloaded into the **/client/codesets** directory on the e*Index workstations as well as into the **/client/bin/codesets** directory in the e*Gate environment. In addition, a new environment variable, INTBIN, must be added to the e*Gate environment in which the e*Index Schemas are located. INTBIN should point to the location of the **/codesets** directory. On the client workstation, the INTBIN variable is automatically created when you install or upgrade the GUI. It is defined as the current working directory (".").

Active Integration Tables

The Java APIs for e*Index Active Integration require that you add one new table to the database, *ui_local_id_generator*. This table enables you to define a range of local IDs to be assigned by your custom programs for each linked system. In addition, you can add a set of tables to the database to define display customizations for each user of your Java applications. These new tables are:

- *ui_active_set*
- *ui_active_function*
- *ui_active_field*
- *ui_active_field_cfg*
- *ui_active_field_fill*

- *ui_active_user_set*

These tables allow you to define display attributes for your custom applications by setting, system, and function. Display attributes include required fields, default values, field labels, and field triggers and their associated fill values. You can customize the appearance of the applications for each user.

For information about installing any of the active integration tables, see chapter 7 of the *e*Index Global Identifier Installation Guide* or the *e*Index Global Identifier Upgrade Guide*. For information about the structure and use of the new tables, see chapter 3 of the *Java Programmer's Reference for e*Index Active Integration*.

Java APIs for e*Index Active Integration

Overview

This release of e*Index adds a set of Java APIs that enable active integration of e*Index with existing systems. If you are already working with active integration APIs for e*Index 4.1.1, this section of the release bulletin outlines the primary differences between versions 4.1.1 and 4.5.3. For detailed information about the changes listed, see the appropriate section in the *Java Programmer's Reference for e*Index Active Integration*.

If you are new to the Java APIs, see the *Java Programmer's Reference for e*Index Active Integration* for detailed information about implementing the classes and methods provided with the API.

Codeset Implementation

To ensure that the Java APIs are using the same set of codesets for Vality data mapping as the e*Index e*Ways and client workstations, the codesets are now downloaded into the **/configs/codesets** directory when you install the APIs. A new environment variable, INTBIN, must be created, and should point to the location of the **/codesets** directory.

New Class Hierarchy

The class hierarchy was modified slightly for this release. A new class, **com.stc.eIndex.active.core.DataObject**, was added and it extends **java.lang.Object**. Direct subclasses of this class are:

- **com.stc.eIndex.active.person.CountryOption** (also a new class)
- **com.stc.eIndex.active.person.DemographicsRO**
- **com.stc.eIndex.active.core.DynamicDataObject** (also a new class)
- **com.stc.eIndex.active.person.EiSystem**
- **com.stc.eIndex.active.person.PersonDependent**
- **com.stc.eIndex.active.person.Transaction**

The **DemographicResultsRO** class extends **DynamicDataObject**. Classes representing auxilliary person information extend the **PersonDependent** class. These classes include:

- **com.stc.eIndex.active.person.Address**
- **com.stc.eIndex.active.person.Alias**
- **com.stc.eIndex.active.person.Audit**

- **com.stc.eIndex.active.person.AuxId**
- **com.stc.eIndex.active.person.LocalId**
- **com.stc.eIndex.active.person.Phone**

New Classes in the Person Package

Several new classes were added to the Person package to accommodate the new database structure and to provide additional display customizations for your Java applications. The new classes also support country-specific attributes. The new classes in the Person package are:

- **ActiveField**
  This class represents information about the fields defined for display customization.
- **ActiveFieldFill**
  This class retrieves an enumeration of field fill triggers and field fill values for the fields in **ActiveField**.
- **ActiveFunction**
  This class represents information about the functions defined for display customization.
- **ActiveLookup**
  This class looks up the information defined for display customization.
- **ActiveSet**
  This class represents information about the settings and systems defined for display customization.
- **Address**
  This class represents an address associated with a person record.
- **Audit**
  This class represents an audit log entry associated with a person record.
- **AuxId**
  This class represents a non-unique ID record associated with a person record.
- **CountryOption**
  This class represents a set of country-specific option data.
- **CountryOptionLookup**
  This class looks up country-specific option definitions.
- **EnumCountryOption**
  This class creates the enumerations used by the **CountryOptionLookup** class to retrieve country-specific option sets.
- **Phone**
  This class represents a telephone record associated with a person record.
- **PredefinedMsg**
  This class represents a predefined message that can be associated with a person record.

- **PredefinedMsgRegistry**
  This class retrieves an enumeration of all available predefined messages.

New Interface in the Person Package

One new interface was added to the Person package to allow you to set fields to null. Previously this was handled by the **setFieldNull** method in the **Demographics** class. Now it is handled by the interface ***EiNull***.

Alias Class

Several changes were made to the **Alias** class to accommodate new functionality. The properties of the **Alias** class were changed so the class extends **com.stc.eIndex.active.person.PersonDependent** instead of **java.lang.Object**.

- The **Alias** class contains three new methods to enable updating an existing alias name in a person record. The new methods are **setLastName**, **setFirstName**, and **setMiddleName**. These methods set the new values for the alias name fields.
- To accommodate the field name change from *middle_initial* to *middle_name*, the **getMiddleInitial** method was changed to **getMiddleName**.
- The **createDate** method was removed since this information is now stored for the transaction record rather than the alias record.

CodeDisplay Class

In e*Index 4.5.3, code table data is stored in one database table instead of each type being stored in its own table. This means religions, races, languages, countries, and so on are all stored in one table, *stc_common_detail*. The table *stc_common_header* stores the codes and descriptions for the different types of data elements stored in *stc_common_detail*. The following changes were made to the **CodeDisplay** class to enable access to the new tables.

- A new method, **getCommonDetailId**, was added to retrieve ID codes for data elements in *stc_common_detail*.
- A new method, **getCommonHeaderId**, was added to retrieve ID codes for data element types *stc_common_header*.
- Two new methods, **getCreateDate** and **getCreateUserId**, were added to retrieve information about when a data element was created and who created it.

- One new method, **getReadOnly**, was added to retrieve an indicator that specifies whether the data element record can be modified.

ControlKey Class

New control keys were added to e*Index after version 4.1.1, and new methods were added to the **ControlKey** class to obtain the value of new control keys and of some existing control keys. The new methods are:

- **getCheckSumLength**
- **isBlankOnUpdateEnabled**
- **isMixedCaseEnabled**
- **isSameFacilityReportEnabled**
- **isShortIdEnabled**

Demographics Class

Several changes were made to the **Demographics** class to accommodate additional data elements and the new database structure. The **Demographics** class now inherits **toString** from **com.stc.eIndex.active.core.DataObject**, and it now has two sub-classes: **com.stc.eIndex.active.person.Person** and **com.stc.eIndex.active.person.SearchParameters**. Several methods were added to the class, and several methods were removed.

New Methods

The following are methods added to the **Demographics** class for this release.

- **setCitizenship**
- **setDate1**
- **setDate2**
- **setDate3**
- **setDate4**
- **setDate5**
- **setDateOfDeath**
- **setDeathCertificate**
- **setDistrictOfResidence**
- **setDriversLicenseNumber**
- **setDriversLicenseState**
- **setLgaCode**
- **setMilitaryBranch**
- **setMilitaryRank**

- **setMilitaryStatus**
- **setNationality**
- **setPensionExpirationDate**
- **setPensionNumber**
- **setPersonCategoryCode**
- **setRepatriationNumber**
- **setString1**
- **setString10**
- **setString2**
- **setString3**
- **setString4**
- **setString5**
- **setString6**
- **setString7**
- **setString8**
- **setString9**

Obsolete Methods

The following methods were removed from the **Demographics** class. Most of these methods are now included in the **Address** or **Phone** class. A new interface, ***EiNull***, was added to update fields with null values, replacing the method **setFieldNull**.

- **setAddressCity**
- **setAddressCountry**
- **setAddressCountyCode**
- **setAddressStreet1**
- **setAddressStreet2**
- **setAddressZip**
- **setAddressZipExt**
- **setBusinessPhone**
- **setFieldNull**
- **setHomePhone**

Modified Methods

The following changes were made to existing methods in the **Demographics** class.

- **setMiddleInitial** was replaced by **setMiddleName**

- The format for the parameter for **setDob** is now of the type **java.sql.Date** instead of **java.lang.String**.

DemographicsResultsRO Class

Several changes were made to the **DemographicsResultRO** class to accommodate additional data elements and the new database structure. New methods were added to the class, and the **DemographicsResultsRO** class now extends **com.stc.eIndex.active.core.DynamicDataObject** instead of **java.lang.Object**.

New Methods

The following are methods added to the **DemographicsResultRO** class for this release. These methods allow you to retrieve a full set of demographic data in a result set.

- **getCitizenship**
- **getClass1**
- **getClass2**
- **getClass3**
- **getClass4**
- **getClass5**
- **getDate1**
- **getDate2**
- **getDate3**
- **getDate4**
- **getDate5**
- **getDateOfDeath**
- **getDeath**
- **getDeathCertificate**
- **getDistrictOfResidence**
- **getDriversLicenseNumber**
- **getDriversLicenseState**
- **getEthnic**
- **getFatherName**
- **getLgaCode**
- **getMaidenName**
- **getMaritalStatus**
- **getMilitaryBranch**
- **getMilitaryRank**

- **getMilitaryStatus**
- **getMotherMaidenName**
- **getNationality**
- **getPensionExpirationDate**
- **getPensionNumber**
- **getPersonCategoryCode**
- **getPobCity**
- **getPobCountry**
- **getPobState**
- **getRace**
- **getReligion**
- **getRepatriationNumber**
- **getSpouseName**
- **getString1**
- **getString10**
- **getString2**
- **getString3**
- **getString4**
- **getString5**
- **getString6**
- **getString7**
- **getString8**
- **getString9**
- **getSuffix**
- **getTitle**
- **getVeteranStatus**
- **getVipFlag**

Obsolete Methods

One method, **getAddressStreet1**, was removed from the **DemographicsResultRO** class since addresses are no longer part of a member's demographic information.

Modified Methods

The following changes were made to existing methods in the **DemographicsResultRO** class.

- **getMiddleInitial** was replaced by **getMiddleName**

- The format for **getDob** is now of the type **java.sql.Date** instead of **java.lang.String**.

DemographicsRO Class

Several changes were made to the **DemographicsRO** class to accommodate additional data elements and the new database structure. New methods were added to the class and obsolete methods were removed. The **DemographicsRO** class now extends **com.stc.eIndex.active.core.DynamicDataObject** instead of **java.lang.Object**.

New Methods

The following are methods added to the **DemographicsRO** class for this release. In addition, the **DemographicsRO** class inherits the **toString** method from **com.stc.eIndex.active.core.DataObject**.

- **getCitizenship**
- **getDate1**
- **getDate2**
- **getDate3**
- **getDate4**
- **getDate5**
- **getDateOfDeath**
- **getDeathCertificate**
- **getDistrictOfResidence**
- **getDriversLicenseNumber**
- **getDriversLicenseState**
- **getLgaCode**
- **getMilitaryBranch**
- **getMilitaryRank**
- **getMilitaryStatus**
- **getNationality**
- **getPensionExpirationDate**
- **getPensionNumber**
- **getPersonCategoryCode**
- **getRepatriationNumber**

- **getString1**
- **getString10**
- **getString2**
- **getString3**
- **getString4**
- **getString5**
- **getString6**
- **getString7**
- **getString8**
- **getString9**

Obsolete Methods

Several methods were removed from the **DemographicsRO** class because of the new database structure. Most of these methods were moved to the new **Address** or **Phone** class.

- **getAddressCity**
- **getAddressCountry**
- **getAddressCountyCode**
- **getAddressState**
- **getAddressStreet1**
- **getAddressStreet2**
- **getAddressZip**
- **getAddressZipExt**
- **getBusinessPhone**
- **getCommaDelimitedDemographics**
- **getHomePhone**
- **toString**

Modified Methods

The following changes were made to existing methods in the **DemographicsRO** class.

- **getMiddleInitial** was replaced by **getMiddleName**
- The format for **getDob** is now of the type **java.sql.Date** instead of **java.lang.String**.

EiBOFactory Class

In the **EiBOFactory** class, the **getPersonBOInstance** method now throws two exceptions: **EiException** and a **java.sql.Exception**.

EiConnection Class

The **EiConnection** class has two new methods: **getEiServer** and **getStartTransactionStatement**. **getStartTransactionStatement** is for internal use only, and you should not need to use it. In the current release, only one parameter is accepted for the **prepareStatement** method. The accepted syntax for this statement is:

```
public java.sql.PreparedStatement
prepareStatement(java.lang.String sqlStatement)
```

EiServer Class

Several new methods were added to the **EiServer** class for this release. The new methods are:

- **equals**
- **getDbTimestamp**
- **getLogicalName**
- **getProperties**
- **hashCode**

In addition, the **EiServer** constructor method now has two possible syntaxes, as shown below.

```
public EiServer(java.lang.String propertyFileName)
```

or

```
public EiServer(java.util.Properties propertyFileName)
```

EiSystem Class

For this release, the following changes were made to the **EiSystem** Class.

- **EiSystem** now extends **com.stc.eIndex.active.core.DataObject** instead of **java.lang.Object**
- Two new methods, **isFormatMatch** and **getRegionCode**, were added.
- The format for **getCreateDate** is now of the type **java.sql.Timestamp** instead of **java.util.Date**.
- The **toString** method was removed, and **EiSystem** now inherits **toString** from **com.stc.eIndex.active.core.DataObject**.

EnumCodeType Class

For this release, several new fields were added to the **EnumCodeType** class to accommodate the new types of code table data. The new fields include:

- **ADDRESS_TYPE**
- **CITIZENSHIP**
- **DISTRICT_OF_RESIDENCE**
- **DRIVER_LICENSE_ISSUER**
- **EVENT_NOTIFICATION**
- **NATIONALITY**
- **PERSON_CATEGORY**
- **PHONE_TYPE**
- **REGION**

EnumDemographicField Class

For this release, the **EnumDemographicField** class was removed from the API. This class was previously used to set demographic fields to null. This is now handled by the interface ***EiNull***.

EnumLocalIdStatus Class

The **getEnumeration** method threw an EiParameterException in the previous version. In the new version, **getEnumeration** throws an EiException.

LocalId Class

The **LocalId** class previously extended **java.lang.Object**; now it extends **com.stc.eIndex.active.person.PersonDependent**. The **LocalId** class now inherits the **toString** method from **com.stc.eIndex.active.core.DataObject**.

LocalIdBO Class

Two new methods were added to the **LocalIdBO** class: **getLocalIdEnumeration** and **lookupLocalId**.

Person Class

Three new methods were added to the **Person** class to enable you to retrieve the address, telephone, and non-unique ID information associated with a person record. The new methods are:

- **getAuxIdEnumeration**

- **getAddressEnumeration**
- **getPhoneEnumeration**

The **toString** method was removed from the **Person** class, which now inherits **toString** from **com.stc.eIndex.active.core.DataObject**.

PersonBO Class

Several new methods were added to the **PersonBO** class to allow you to add new address, telephone, non-unique ID, audit, and comment information to person records. Methods were also added to accommodate the expanded search capabilities of e*Index. The new methods are:

- **addAddress**
- **addAudit**
- **addAuxId**
- **addPhone**
- **addPredefinedComment**
- **addUserDefinedComment**
- **hasAddress**
- **hasAuxId**
- **hasPhone**
- **searchGeneral**
- **uidLookupByAuxId**

SearchParameters Class

The following changes were made to the **SearchParameters** class for this release.

- Previously, the **SearchParameters** class extended **java.lang.Object**. In the new release, the **SearchParameters** class extends **com.stc.eIndex.active.person.Demographics** and inherits all methods from the **Demographics** and **DemographicsRO** classes.
- **SearchParameters** inherits **toString** from **com.stc.eIndex.active.core.DataObject**.
- All of the methods for **SearchParameters** that existed for version 4.1.1 were removed for this version. Those methods are now inherited from the **Demographics** class.
- One new method, **setUid**, was added to allow you to set the UID as a search parameter.
- The **SearchParameters** constructor method now throws an **EiException**.

Security Class

The **Security** class no longer checks the obsolete database table *stc_user* to verify whether a user ID is valid. It checks the new database table *user_tbl*.

Ssn Class

The **Ssn** class has one new field, **NULL**. This field allows you to set fields of the type **Ssn** to null.

Transaction Class

For this release, the following changes were made to the **Transaction** class.

- The **Transaction** class no longer extends **java.lang.Object**; now it extends **com.stc.eIndex.active.core.DataObject**.
- One new method, **equals**, was added to the class.
- The **Transaction** constructor method no longer takes *function* as a parameter. The function of a transaction is populated after a transaction occurs.
- The **Transaction** constructor method no longer throws an EiDataException; it throws an EiException.

Uid Class

The **Uid** class has one new field, **NULL**. This field allows you to set fields of the type **Uid** to null.

Instantiated Classes

Three of the new classes added to the Person package need to be instantiated before their methods can be used. These classes are **ActiveLookup**, **CountryOptionLookup**, and **PredefinedMsgRegistry**.

Exception Package

Several new exception classes were added for this release, and two exception classes were removed.

New Exception Classes

The new exception classes cover a wider variety of error possibilities and each class defines a more specific error type. The new classes are enclosed by the primary exception class, **EiException**, excluding **EiRuntimeException**. Each class contains one constructor method. For detailed information about the new exception classes, refer to chapter 5 of the *Java Programmer's Reference for e*Index Active Integration*. The new exception classes are listed below.

- **EiException.DataNotFound**
- **EiException.DeleteRow**
- **EiException.EmptyString**
- **EiException.ExceedMaxStringLength**
- **EiException.GeneralException**
- **EiException.InvalidData**
- **EiException.InvalidHistory**
- **EiException.InvalidLocalIdStatus**
- **EiException.InvalidParameter**
- **EiException.InvalidPersonStatus**
- **EiException.InvalidTimestamp**
- **EiException.InvalidUid**
- **EiException.InvalidUniqueKeyQuery**
- **EiException.MatchException**
- **EiException.NonUpdateableFieldModified**
- **EiException.RequiredFieldIsNull,**
- **EiException.UniqueKeyException**
- **EiException.UpdateRow**
- **EiRuntimeException**

Obsolete Exception Classes

Two exception classes were removed for this release. The functions of the obsolete classes can be performed using some of the new exception classes listed above.

- **EiDataException**
- **EiParameterException**

Log Files

The naming convention for the log files that are created when the Java APIs are run has changed for this release. You can specify the name for the log files in the file **EiServer.properties**. This file has a new variable named **logicalName**. Each instance of EiServer must have a unique logical name. Only one log file is created for each logical name, and information is appended to the log file for each JVM session.