Maintaining Sun Master Indexes

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 821–0863–10
December 2009

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

091220@23031

Contents

Maintaining Sun Master Indexes

The topics listed here provide procedures, conceptual information, and reference information for maintaining a Sun Master Index application.

What You Need to Know

These topics provide information you should know about maintenance tasks.

What You Need to Do

These topics provide instructions on how to perform maintenance tasks.

More Information

These topics provide additional information you should know about maintaining a master index application

Related Topics

Several topics provide information and instructions for implementing and using a master index application. For a complete list of topics related to working with Sun Master Index, see "Related Topics" in *Developing Sun Master Indexes* .

Defining Master Index Data Manager Security

Sun Master Index supports security for the Master Index Data Manager (MIDM) at the user and function level and also supports Secure Sockets Layer (SSL) authentication. Security is defined at two levels, the EJB level and the presentation level. EJB security provides access at the user and function level to the methods of the master controller (`com.sun.mdm.index.ejb.master`). Presentation level security provides access at the function and user level for the actions that can be performed from the MIDM.

A secure user name and password needs to be defined for each master index application user to connect to the database and to log on to the MIDM. For each user account you define, you must specify one or more roles in order for that user to be able to perform any functions in the MIDM. You define roles in `midm-security.xml` in the master index project. This is the presentation layer security. In addition, each user must also be assigned at least one EJB security role. EJB security roles are defined in `security.xml`. A default role that grant access to all functions of the master controller is predefined, but is not included in the file. The role is named MasterIndex.Admin.

User permissions for master index applications are granted using the Admin Console. You can also define security using a Lightweight Directory Access Protocol (LDAP) server, using the roles you define in "Define Master Index Data Manager User Roles" on page 7.

Perform the following tasks to configure security for the master index application:

- "Define Master Index Data Manager User Roles" on page 7
- "Define EJB User Roles" on page 7
- "Create Master Index Data Manager User Accounts" on page 8

These topics provide additional information to help you perform the above tasks:

- "Master Index Data Manager User Role Properties" on page 9
- "Master Index Data Manager User Permissions" on page 10
- "EJB User Role Properties" on page 12
- "EJB Security Functions" on page 13

Define Master Index Data Manager User Roles

Sun Master Index provides sample user roles for giving multiple permissions to a user at one time. You can define additional user roles and assign combinations of access permissions to each role. This way you can assign a user account to one or two user roles instead of assigning them several access permissions.

▼ To Define a User Role

1. **In the NetBeans Project window, expand the master index project and then expand Configuration.**
2. **Open `midm-security.xml` in an XML editor.**
3. **Define user groups and their permissions using the elements described in "Master Index Data Manager User Role Properties" on page 9.**

   The permissions you can assign are listed and described in "Master Index Data Manager User Permissions" on page 10.
4. **Save and close the file.**

   You can use these roles when you create the user accounts, as described in "Create Master Index Data Manager User Accounts" on page 8.
5. **Continue to "Define EJB User Roles" on page 7.**

Define EJB User Roles

EJB user roles control access at the master controller level. Sun Master Index provides a sample role for granting multiple permissions at one time without giving access to all functions. An additional role is predefined, MasterIndex.Admin, that provides access to all functions. You can define additional roles and assign combinations of functional permissions to each role. This way you can assign a user account to one or two roles instead of assigning them several permissions.

Note – This step is optional. You can use the MasterIndex.Admin role for MIDM users if you only need to restrict access at the presentation level.

▼ To Define an EJB User Role

1. **In the NetBeans Projects window, expand the master index project and then expand Configuration.**

2. **Open `security.xml` in an XML editor.**

3. **Define user roles and the permissions that belong to each using the elements described in "EJB User Role Properties" on page 12.**

   The permissions you can assign are listed and described in "EJB Security Functions" on page 13.

4. **Save and close the file.**

   You can use these roles when you create the user accounts.

5. **Continue to "Create Master Index Data Manager User Accounts" on page 8.**

Create Master Index Data Manager User Accounts

You create user accounts for MIDM access using the Sun Java System Application Server Admin Console.

Tip – Make sure you give users access to the initial page that appears when a user logs in to the MIDM. This page is defined in midm.xml. Also verify that the EJB privileges you assign allow the user to perform all of the MIDM functions to which they have access.

▼ To Create a User Account

Before You Begin Make sure you have created all the user roles and, optionally, EJB user roles that need to be assigned to the user account.

1. **Log on to the Sun Java System Application Server Admin Console.**

2. **In the left portion of the page, expand Configuration, expand Security, and then expand Realms.**

3. **Select File.**

4. **On the Edit Realm page, select Manage Users.**

5. **On the File Users page, select New.**

6. **In the User ID field, enter a name for the user.**

7. **In the Group List field, enter the following. Separate roles with a comma.**

   - **`MasterIndex.Admin`** or one or more of the EJB user roles you defined in "Define EJB User Roles" on page 7.

- One or more of the user roles you defined in "Define Master Index Data Manager User Roles" on page 7.

8 **After you have added all required user roles and EJB user roles, enter a password for the user in the New Password field.**

9 **In the Confirm New Password field, enter the password again.**

10 **Click OK.**

Master Index Data Manager User Role Properties

You can define user roles for the MIDM in order to assign multiple security permissions to a user account at once. Roles are defined in an XML file, midm-security.xml. The following table describes the elements of the security configuration file.

TABLE 1 MIDM User Role Configuration Elements

Element	Description
role	A definition for one user role. Each role element contains a name for the user role, a list of security permissions, and, optionally, a user role from which permissions are inherited along with any exceptions to the inheritance.
role-name	The name of the user role, such as Administrator.
inheritance	A definition of how permissions are inherited from another user role. The definition includes the parent user role and any permissions that should not be inherited. This group of elements is optional, and a role can inherit from multiple user roles. **Note –** The role from which permissions are inherited must be defined earlier in the XML file than the role that inherits the permissions.
inherits-from	The name of the user role from which the current role inherits permissions. If permissions are added to this user role at any time, the new permissions are also inherited by the current role.
excluded-operations	A list of permissions assigned to the parent role that the current role should not have access to. Any permissions assigned to the parent role that are not listed here are assigned to the current role. **Note –** If a role inherits from multiple parent roles and each parent is assigned an excluded permission, you need to specify that the permission be excluded for each parent role.

TABLE 1 MIDM User Role Configuration Elements *(Continued)*

Element	Description
excluded-operations/name	The name of a security permission that is not inherited from the parent user role. Security permissions are listed under "Master Index Data Manager User Permissions" on page 10.
operation	A list of security permissions to assign to the user role. If the role inherits permissions from another role, the permissions listed here are in addition to the inherited permissions.
operation/name	The name of a security permission to add to the current user role. Security permissions are listed under "Master Index Data Manager User Permissions" on page 10.

Master Index Data Manager User Permissions

The following table lists and describes each user permission for the MIDM. The user permission names are case-sensitive.

TABLE 2 MIDM User Permissions and Descriptions

User Permission	Description
AssumedMatch_Print	Gives access permission to print the results of an assumed match search.
AssumedMatch_SearchView	Gives access permission to search for and view records that were automatically matched by the master index application. This permission is needed to perform any assumed match functions.
AssumedMatch_Undo	Give access permission to reverse an assumed match, separating the two records.
AuditLog_Print	Gives access permission to print an audit log search results report. This permission also requires AuditLog_SearchView.
AuditLog_SearchView	Gives access permission to search for and view audit log entries.
EO_Activate	Gives access permission to activate enterprise records.
EO_Compare	Gives access permission to compare enterprise records.
EO_Create	Gives access permission to create new enterprise records.
EO_Deactivate	Gives access permission to deactivate enterprise records.
EO_Edit	Gives access permission to modify the SBR in enterprise records.
EO_LinkSBRFields	Gives access permission to link a field in a system record with a field in the enterprise record's SBR so the value of the SBR field is the same value as the system object field.

TABLE 2 MIDM User Permissions and Descriptions *(Continued)*

User Permission	Description
EO_LockSBRFields	Give access permission to modify the SBR directly and to lock SBR fields for overwrite.
EO_Merge	Gives access permission to merge enterprise records.
EO_OverwriteSBR	Gives access permission to choose an SBR field to retain during a merge. After the merge transaction, the field is locked for editing.
EO_PrintComparison	Reserved for future functionality.
EO_PrintSBR	Reserved for future functionality.
EO_SearchViewSBR	Gives access permission to search for and view single best records, and to generate and print the search results report. This permission is needed to perform any functions on the details page.
EO_UnlinkSBRFields	Gives access permission to unlink an SBR field and system record field that were previously linked.
EO_UnlockSBRFields	Gives access permission to unlock an SBR field that was previously locked for editing.
EO_Unmerge	Gives access permission to unmerge enterprise records.
EO_ViewMergeTree	Gives access permission to view a merge history of an enterprise object.
Field_VIP	Gives permission to view fields masked by any custom masking logic specified by midm.xml.
PotDup_Print	Gives permission to print the results of a potential duplicate search.
PotDup_ResolvePermanently	Gives access permission to permanently resolve potential duplicate records.
PotDup_ResolveUntilRecalc	Gives access permission to resolve potential duplicate records.
PotDup_SearchView	Gives access permission to search for and view potential duplicate records. This permission is needed in order to perform any functions on the Duplicate Records page.
PotDup_Unresolve	Gives access permission to unresolve potential duplicate records that were previously resolved.
Reports_Activity	Gives access permission to run an activity report.
Reports_AssumedMatches	Gives access permission to run an assumed match report.
Reports_DeactivatedEUIDs	Gives access permission to run a deactivated record report.
Reports_Duplicates	Gives access permission to run a potential duplicate report.
Reports_MergedRecords	Gives access permission to run a merge transaction report.
Reports_UnmergedRecords	Gives access permission to run an unmerge transaction report.

TABLE 2 MIDM User Permissions and Descriptions *(Continued)*

User Permission	Description
Reports_Updates	Gives access permission to run an update report.
Reports_View	Gives access permission to the reports page. This permission is needed in order to run any of the production or activity reports.
SO_Activate	Gives access permission to reactivate a deactivated system record.
SO_Add	Gives access permission to add system records.
SO_Compare	Gives access permission to compare system records.
SO_Edit	Gives access permission to modify system records.
SO_Deactivate	Gives access permission to deactivate system records.
SO_Merge	Gives access permission to merge system records.
SO_Print	Gives access permission to print the results of a system record search.
SO_Remove	Gives access permission to delete system records.
SO_SearchView	Gives access permission to search for and view system records.
SO_Unmerge	Gives access permission to unmerge system records.
TransLog_Print	Gives permission to print the results of a transaction history search.
TransLog_SearchView	Gives access permission to search for and view the transaction history of enterprise records and to view merged records.

EJB User Role Properties

You can define access roles for the EJB layer in order to assign multiple security permissions to a user or web client at once. EJB roles can be used to secure MIDM users and other clients accessing the master index application, such as web services. Roles are defined in an XML file, `security.xml`. The following table describes the elements of the security configuration file. The default user, MasterIndex.Admin, is not defined in this file, but it gives access to all functions.

TABLE 3 EJB User Role Configuration Elements

Element	Description
ejbSecurity	An indicator of whether EJB security is enabled. Enter **ON** to enable web service security; enter **OFF** to disable web service security.
role	A definition for one EJB user role. Each role element contains a name for the user role and a list of security permissions.
role-name	The name of the EJB user role, such as DataProcessor.

TABLE 3 EJB User Role Configuration Elements *(Continued)*

Element	Description
operation	A list of master controller functions to assign to the user role.
name	The name of a master controller function to add to the current user role. Functions are listed under "EJB Security Functions" on page 13.

EJB Security Functions

The following table lists and describes each security function in the master controller. The permission names are case-sensitive. For more information about these functions, see the Javadocs provided with Sun Master Index. These functions are defined in `com.sun.mdm.index.ejb.master.MasterController`.

TABLE 4 EJB Security Functions and Descriptions

User Permission	Description
activateEnterpriseObject	Gives access permission to change the status of a deactivated enterprise object back to active.
activateSystemObject	Gives access permission to change the status of a deactivated system object back to active.
addSystemObject	Give access permission to add a system object to an enterprise object.
calculatePotentialDuplicates	Gives access permission to calculate potential duplicates for a transaction.
calculateSBR	Gives access permission to calculate a new single best record (SBR) for an enterprise object that has been updated.
createEnterpriseObject	Gives access permission to create a new enterprise object in the master index application.
deactivateEnterpriseObject	Gives access permission to change the status of an enterprise object to inactive.
deactivateSystemObject	Gives access permission to change the status of a system object to inactive.
deleteSystemObject	Gives access permission to delete a system object from an enterprise object.
executeMatch	Gives access permission to process a system object using the standardization and matching logic defined for the master index application.
executeMatchDupRecalc	Gives access permission to process a system object using the standardization and matching logic defined for the master index application and allows you to defer potential duplicate processing.
executeMatchGui	Gives access permission to process a system object using the standardization and matching logic defined for the master index application.

TABLE 4 EJB Security Functions and Descriptions *(Continued)*

User Permission	Description
executeMatchUpdate	Gives access permission to process a system object using the standardization and matching logic defined for the master index application.
executeMatchUpdateDupRecalc	Gives access permission to process a system object using the standardization and matching logic defined for the master index application and allows you to defer potential duplicate processing.
getConfigurationValue	Gives access permission to retrieve the configuration of a master controller parameter.
getDatabaseStatus	Give access permission to retrieve the status of the master index database.
getEnterpriseObject	Gives access permission to retrieve an enterprise object.
getEUID	Gives access permission to retrieve the EUID associated with a system and local ID.
getMergeHistory	Gives access permission to retrieve a tree structure of the merge transactions associated with a specific enterprise object.
getRevisionNumber	Gives access permission to retrieve the SBR revision number for an enterprise object.
getSBR	Gives access permission to retrieve the SBR for an enterprise object.
getSystemObject	Gives access permission to retrieve a system object based on the system and local ID information.
insertAuditLog	Gives access permission to add an audit log record to the master index database.
lookupAssumedMatches	Gives access permission to retrieve a list of assumed matches based on the search criteria specified.
lookupAuditLog	Gives access permission to retrieve an audit log record.
lookupPotentialDuplicates	Gives permission to retrieve a list of potential duplicate records.
lookupSystemDefinition	Gives permission to retrieve the attributes of a source system in the master index database.
lookupSystemDefinitions	Gives access permission retrieve the attributes of multiple source systems in the master index database.
lookupSystemObjectPKs	Gives access permission to retrieve an array of system object keys.
lookupSystemObjects	Gives access permission to retrieve the active system objects in an enterprise object.
lookupTransaction	Gives access permission to retrieve a transaction summary.
lookupTransactions	Gives access permission to retrieve an array of transaction summaries.

TABLE 4 EJB Security Functions and Descriptions *(Continued)*

User Permission	Description
mergeEnterpriseObject	Gives access permission to merge two or more enterprise objects.
mergeSystemObject	Gives access permission to merge two or more system objects.
ResolvePotentialDuplicates	Gives access permission to flag a potential duplicate pair as resolved.
searchEnterpriseObject	Gives access permission to retrieve an iterator of enterprise objects based on the specified search criteria.
transferSystemObject	Gives access permission to transfer a system object from its current enterprise object to a different enterprise object.
UndoAssumedMatch	Gives access permission to reverse an assumed match transaction, unmerging the two objects that were matched and creating a new enterprise object.
unmergeEnterpriseObject	Gives access permission to unmerge two previously merged enterprise objects.
unmergeSystemObject	Gives access permission to unmerge two previously merged system objects.
unresolvePotentialDuplicate	Gives access permission to mark as unresolved two potential duplicate records that were previously flagged as resolved.
updateEnterpriseDupRecalc	Gives access permission to update the master index database to reflect new values for an enterprise object and optionally to defer potential duplicate processing.
updateEnterpriseObject	Gives access permission to modify enterprise objects.
updateSystemObject	Gives access permission to modify system objects

Learning About Master Index Reports

Several standard reports are provided with master index applications that allow you to monitor and review the state of the information in the master index database. You can either run these reports through the MIDM or from a command line. The following topics provide an overview of each report.

Master Index Command Line Reports

Sun Master Index provides a set of production and activity reports that can be generated from a command line or from the MIDM. The command line report client is created in *NetBeans_Projects**Project_Name*\`report-client` when you generate the master index application.

The production reports provide information about transactional changes to the data in the master index application and about the current state of that data, helping you monitor stored data and determine how that data needs to be updated. This information also helps verify that the matching logic and weight thresholds are defined correctly. Activity reports provide statistical information for transactions over specific periods of time.

In order to run the command line reports, you must have the Java Runtime Environment (JRE) 1.5.13 or later installed on the machine where the report files reside. For additional reporting needs, the database is accessible using any commercially available ODBC-compliant reporting tool. You can also define reports using Java, PL/SQL, or SQL.

About Production Reports

Production reports should be run daily and provide information about the transactions that are processed through the master index database. These reports provide lists of potential duplicate records, merge transactions, unmerge transactions, assumed matches, updates, and deactivated records for a specified time period. The information you find in these reports helps you analyze your matching threshold configuration, and provides valuable information about how data is being processed with your current configuration. In addition to running the production reports daily, you should run them against any data that has been loaded from existing systems into the master index database in batch format.

About Activity Reports

Activity reports should be run weekly, monthly, and yearly to obtain statistical data about the transactions that are processed through the master index database. These reports give the number of each type of transaction performed for the specified week, month, or year. They also provide cumulative information for the week, month, or year to date. The information you find in these reports helps analyze the matching threshold configuration and the condition of your data by giving you the number of potential duplicates created, the number of assumed matches, and so on.

Master Index Report Configuration

The reports are configured by XML files. For the command line reports, the configuration files are located in the report home directory in the `config` subdirectory. The file `CompanyReport.xml` provides an example of how the file might be configured for a company

object; the file `PersonReport.xml` provides an example of how the file might be configured for a person object. You can use either file for your reports. When you create a new master index application, you can specify the fields that appear on reports.

The configuration files allow you to specify which reports to run, the time period of the transactions to include in each report, and the name and location of the report files. You can also define various report details, such as the name of each report, which fields to include, and the names and sizes of the report columns. Most of these changes should only need to be made one time, before you first run the reports.

Creating Custom Master Index Reports

If the standard reports do not provide you with all the information you need, you can create custom reports using PL/SQL, SQL, or Java (using the "lookup" methods in the MasterController class). You can also access the database using any ODBC-compliant report writer (such as Crystal Reports), providing you with the flexibility to report on any information contained in the master index database.

Masked Data in Master Index Reports

The MIDM can be configured to hide certain fields from users who do not have the appropriate security permissions. However, reports will display hidden data if those fields are configured to appear on the reports. Be sure to only give access to users who should be able to view this information, or do not include hidden fields in the reports.

Master Index Production Reports

The standard production reports help you to monitor and analyze the data in the master index database. You can view information about the transactions processed and about any potential duplicates or assumed matches that result from these transactions.

Each report has certain fields that are always displayed and certain fields that are configured to display. You can customize the configured fields that appear on each report as needed. By default, `CompanyReport.xml` configures all reports to include the company name, type, stock symbol, primary contact, street address, city, and telephone number fields. `PersonReport.xml` configures all reports to include the first name, last name, date of birth, SSN, and address line 1 and 2 fields. The fields that are always displayed are described for each report in the following sections.

Production reports can be run for the current day, the previous day, or for a date range you specify. If you run your daily reports in the evening, you should run the current day's reports. If you run your daily reports in the morning, you should run the previous day's reports.

Assumed Match Report

This report displays information about any records that were automatically updated by incoming data during the specified time period. The information in this report, in combination with data from the potential duplicate report, helps you determine whether the matching threshold for assumed matches is accurate. You should review this report daily to ensure that no assumed matches were made in error. The master index application provides the ability to undo an assumed match that was made in error.

The assumed match report always includes the following information about the record that was updated: enterprise-wide unique identifier (EUID), system code, local ID, and matching weight. The report provides the same information for the incoming message that updated the existing record with the exception of the EUID. You can configure the report to include any additional fields from the defined object structure in object.xml in the master index project.

Deactivated Record Report

This report displays a list of all enterprise records that were deactivated during the specified time period. This report does not include system records that were deactivated. Review this report daily to ensure that no records were deactivated in error. The master index application provides the ability to reactivate any deactivated record. The deactivated record report always includes the EUID of the deactivated record, and you can configure the report to include any additional fields from the defined object structure in object.xml in the master index project.

Potential Duplicate Report

This report displays information about records that were marked as potential duplicates of one another during the specified time period. The information provided on this report can help you determine whether the matching (or upper) threshold and the duplicate threshold are configured accurately. The information for each record on the potential duplicate report always includes the EUIDs of both records, the system code, and the matching weight between each potential duplicate pair. You can configure the report to include any additional fields from the defined object structure in object.xml in the master index project.

If same system matching is not enabled and two duplicate records from the same system on this report have a matching weight above the match threshold, it is an indication that the records most likely represent the same person. Review the potential duplicate report daily to determine if two records need to be merged or if they can be resolved. Use this report as a work list when working with potential duplicates.

Merge Transaction Report

This report displays a list of all enterprise records that were merged during the specified time period. Review this report daily to ensure that no records were merged in error. The master index application provides the ability to unmerge any merged records. The merge transaction

report always includes the EUID of each record affected by the merge. You can also configure the report to include any additional fields from the defined object structure in object.xml in the master index project.

UnMerge Transaction Report

This report displays a list of all enterprise records that were unmerged during the specified time period. This report always includes the EUIDs of both records involved in the unmerge transaction, and you can configure the report to include any additional fields from the defined object structure in object.xml in the master index project.

Update Report

This report displays records whose information was updated during the specified time period. Review this report daily to verify the updates made in a given day. This report can help explain why a resolved potential duplicate listing was reinstated to the potential duplicate list. The update report always includes the following information about the record that was updated: EUID, system code, and local ID. You can configure the report to include any additional fields from the defined object structure in object.xml in the master index project. The updated fields might not necessarily appear on this report.

Master Index Activity Reports

The activity reports help you to monitor and analyze the transactions in the master index database by providing statistical data about each transaction type. Unlike the production reports, the information displayed on the activity reports is not configurable. The information displayed on these reports is described for each report in the following sections. Activity reports can be run for any week, month, or year you specify.

Weekly Activity Report

This report displays a summary of transactions that occurred against the database on each day for the specified calendar week (always Sunday through Saturday). The information provided in this summary includes the number of each of the following transactions performed each day.

- Add
- Update
- EUID Deactivate
- EUID Merge
- EUID Unmerge
- LID Merge
- LID Unmerge
- LID Transfer

Monthly Activity Report

This report displays a summary of transactions that occurred against the database during the specified month. You can run this report for any calendar month. The information provided in this summary includes the number of each of the following transactions that were performed for the month:

- Add
- EUID Deactivate
- EUID Merge
- EUID Unmerge
- LID Merge
- LID Unmerge
- Unresolved Potential Duplicates
- Resolved Potential Duplicates

Yearly Activity Report

This report displays a summary of transactions that occurred against the database for the specified calendar year. You can run this report for any calendar year. The information provided in this report includes a summary of each transaction listed for the monthly activity report above.

Master Index Database Indexes

Some of the reports you run can grow quite large, impacting the performance of the report client. The following indexes are created in the database to improve performance.

```
CREATE INDEX SBYN_POTENTIALDUPLICATES3 ON SBYN_POTENTIALDUPLICATES
 (TRANSACTIONNUMBER ASC);

CREATE INDEX SBYN_ASSUMEDMATCH2 ON SBYN_ASSUMEDMATCH (TRANSACTIONNUMBER ASC);

CREATE INDEX SBYN_TRANSACTION4 on SBYN_TRANSACTION (EUID2 ASC, TIMESTAMP ASC);

CREATE INDEX SBYN_TRANSACTION3 on SBYN_TRANSACTION (TIMESTAMP ASC,
 TRANSACTIONNUMBER ASC);
```

Note – These indexes should be removed prior to performing an initial load or batch load of data.

Working With Master Index Command Line Reports

The following topics provide information and instructions for configuring and running reports from a command line using a Java command. For information about running the reports from the Master Index Data Manager, see *Working With the Master Index Data Manager*.

The reports are automatically generated at *NetBeans_Projects*/*Project_Name*/`report-client`. You must also have the Java 2 Platform, Standard Edition v. 1.5.13 or later installed on the machine from which the reports are run. Be sure you have configured the database connection for the master index application using the Sun Java System Application Service Admin Console.

Configuring the Master Index Report Environment

Before running the master index reports from a command line, you must configure the report environment.

▼ To Set up the Environment

1. **If you install or move the reports files to a machine other than the application server machine, make sure JRE 1.5.13 or later is installed on the machine where the files reside.**
2. **Set up all Java environment variables as specified in the Java documentation.**
3. **Create one environment variable, JAVA_HOME, and set it to the home directory of the JRE installation.**
4. **If you run the reports using the Java command and not the supplied batch file, modify the CLASSPATH variable before running the reports for the first time by adding the absolute path and filename of the files in the `lib` subdirectory of the reports home directory to the CLASSPATH variable.**

Configuring Master Index Command Line Reports

Before running any reports from the command line, you must customize the XML configuration file. You can use either of the files located in the reports directory in the eView or eIndex subdirectory. A default XML file named `PersonReport.xml` is defined for a person object and a default XML file named `CompanyReport.xml` is defined for a company object. You

can use either of these as a basis for your production configuration file. Report configuration includes two steps: defining the overall report configuration and configuring the individual reports.

Defining the Command Line Report Configuration

The first section of the report configuration file is indicated by the *DOCTYPE* and the *report* elements and tells the report client how to connect to the application server, which application to run the reports against, and where to output the report files.

Note – The DOCTYPE element indicates the type of document being generated. Do not change this value.

▼ To Define the Command Line Report Configuration

1. **In the SYSTEM element, enter the location of the DTD file for the reports.**

   By default, this file is named `report.dtd`, and is located in the `config` directory. You should not need to modify this attribute unless you move `report.dtd`

2. **In the *appserver* element, enter the IIOP address for the application server.**

   This must be in the format `corbaname:iiop:`*host*`:`*port*, where *host* is the name of the server and *port* is the ORB port number.

3. **In the *application* element, enter the name of the primary object used by the master index application.**

4. **In the *output-folder* element, enter the location in which the generated reports will be placed.**

   If an output directory is specified in the command line, that directory overrides the one specified here. If the output directory already exists, the report client issues a warning that any existing report files will be overwritten and gives you the option of cancelling the reports.

Configuring Command Line Reports

A configuration section is defined for each of the six report templates. Use these sections to configure each report to display information as you want to view it. You can also specify which reports to run.

▼ To Configure Command Line Reports

For each report, make the following modifications before running the reports. Each element or attribute mentioned in the following instructions is defined in . There are six stanzas for you to modify, one for each report.

1. **In the XML file you will use for your implementation, scroll to the *report* element.**

2. **Name the report in the report *name* attribute.**
3. **Specify whether or not to run the report in the *enable* element.**
4. **Define the name of the output file in the *output-file* element.**
5. **Specify a time period for the report by modifying the *type* element and, optionally, the *from-date* and *to-date* elements.**
6. **Define the fields to include on the report by modifying the elements in the *fields* element.**
7. **When you have finished configuring each report, save and close the file.**

   A sample report configuration appears below.

```
<report name="Potential Duplicate Today"
 template="Potential Duplicate">
  <enable>true</enable>
  <output-file>pot_dup_t.txt</output-file>
  <max-result-size>0</max-result-size>
  <page-size>100</page-size>
  <criteria>
    <dates type="today" from-date="" to-date=""/>
    <status></status>
  </criteria>
  <fields>
    <field path="Person.FirstName" label="First Name" width="10"/>
    <field path="Person.LastName" label="Last Name" width="10"/>
    <field path="Person.SSN" label="SSN" width="9"/>
    <field path="Person.DOB" label="DOB" width="10"/>
    <field path="Person.Address.AddressLine1"
      label="AddressLine1" width="30"/>
    <field path="Person.Address.AddressLine2"
      label="AddressLine2" width="30"/>
   </fields>
</report>
```

Master Index Command Line Report Properties

The following table lists and describes the elements in the report configuration files that define the configuration of each production and activity report.

Element/Attribute	Description
report	Defines each report run by the batch file. Each report is defined by a report element.
report/name	The descriptive name of the report. This can be any string, and appears as the title in the specified report.
report/template	The template to use for the type of report being generated. You should not need to modify this element, but you can specify any of the following templates. ■ Assumed Match ■ Potential Duplicate ■ Deactivated ■ Merged ■ Unmerged ■ Update ■ Weekly Activity ■ Monthly Activity ■ Yearly Activity
enable	Specifies whether to run the report for the current run. Specify **`true`** to run the report; specify **`false`** to disable the report. This option allows you to run one report at a time.
output-file	The name of the file generated by the report client. This file is created in the output directory defined earlier in the file or in the output directory specified in the command line (the command line output directory overrides the configuration file output directory).
max-result-size	The number of records to display on the report. If no value is entered, or if the value is zero (0), the size defaults to 1000 records. To retrieve all records for a report, enter a very large value for this element.
page-size	The number of records returned to the report generator at one time for each report.
criteria	Defines the date range for the report.
dates/type	Indicates the type of date range to use for the report. Specify **`today`** to report on transactions with today's date; specify **`yesterday`** to report on transactions with yesterday's date; or specify **`range`** to enter a specific range of dates. If you specify **`range`**, you must enter the date range in the *from-date* and *to-date* attributes. **Note –** If you enter a type of today or yesterday and you enter a date range, only the type will be used. For the activity reports, entering today runs the report for the current week, month, or year. Entering yesterday only runs the previous week's report if yesterday was a Saturday.

Element/Attribute	Description
dates/from-date	The starting date when using a date range for the report. Enter the starting date for the report transactions in **YYYYMMDD** or **YYYYMMDDHHmmss** format. If you enter a date in this element, you must enter a later date in the *to-date* element and specify **range** in the *type* element. **Note –** For the activity reports, you can enter the range for the week, month, or year (depending on the type of activity report) on which you want to report. If the dates you specify do not fall within one calendar week, month, or year, the report client creates a report for the calendar week, month, or year containing the `from-date` and ignores the `to-date` value.
dates/to-date	The ending date when using a date range for the report. Enter the ending date for the report in **YYYYMMDD** or **YYYYMMDDHHmmss** format.
status	This element is valid for the potential duplicate report only, and indicates the status of the potential duplicate pairs to display on the report. Specify any of the following values: ■ **U** - Only unresolved potential duplicates appear on the report. ■ **A** - Only potential duplicates that are permanently resolved (auto-resolved) appear on the report. ■ **R** - Only resolved potential duplicates appear on the report. Leaving the status blank results in potential duplicates of all statuses appearing on the report.
fields	A list of fields to display on the report in addition to those that are displayed automatically. This element should be empty for the activity reports. If a list of fields is supplied for these reports, it is ignored.
field/path	The ePath to a field you want to include in the report. For more information about ePaths, see "Master Index Field Notations" in *Understanding Sun Master Index Configuration Options* . **Note –** You cannot use the asterisk option in the ePaths you specify here.
field/label	The column label for the specified field in the report.
field/width	The width of the column for the specified field in the report. If a field value is larger than the width specified, that value will be truncated in the report.

Running Master Index Command Line Reports

Once you have configured the reports, you can run them by either running the batch file provided with the reports or using the Java command.

Caution – The application server must be running with the master index project deployed and enabled in order to generate command line reports.

To Run the Reports Using the Batch File

1. **From a command prompt, navigate to the location of the report files.**

2. **Type the following all on one line:**

   `ReportClient.bat -f` *config_file*`- d` *output_directory*

   where *config_file* is the name of the report configuration file to use, and *output_directory* is the location to which the reports will be written. This value overwrites the value specified in the configuration file. If this option is not specified, the configuration file value is used.

Note – The `ReportClient.bat` file must reside in the reports home directory at the same level as the `lib` and `config` subdirectories in order for the environment variables to be set up correctly.

3. **To view the reports, navigate to the location you specified as your output path and open the files in any text editor.**

To Run the Reports Using a Java Command

Before You Begin Before running the reports for the first time, set up the environment variables as described in "To Set up the Environment" on page 21.

1. **At the command prompt, type the following all on one line:**

   `java com.sun.mdm.index.report.ReportClient- f` *config_file*`- d` *output_directory*

   where *config_file* is the name of the report configuration file to use and *output_directory* is the location to which the reports will be written. This value overwrites the value specified in the configuration file. If this option is not specified, the configuration file value is used.

Note – An additional option, `-h`, can be used to obtain help information for the report client.

2. **To view the reports, navigate to the location you specified as your output path and open the files in any text editor.**

Maintaining the Master Index Database

The database requires periodic maintenance tasks, such as backing up information or archiving certain tables. Perform backups regularly, and use the standards and policies of your organization to determine the best methods for backing up data. The following topics provide information about tasks you should perform for standard database maintenance.

Backing up the Master Index Database

The master index database must be backed up on a regular basis. Typically, the database should be backed up once a month or once a quarter, depending on the size of the database and the volume of data being processed. The frequency of your database backups depends on your organization's internal policies and practices. Use your normal procedures for backing up a high availability database (this procedure should be determined by a database administrator).

Online Backups

The best practice for backing up the master index database is an online backup during which the database is not shut down. (Note that this does require an offline backup as a starting point to which any online changes can be applied in the event the database must be restored). An online backup will always take a consistent snapshot, though it might not backup all transactions in progress.

Each transaction in the master index application is saved under one commit command, so the state of the database is always consistent when a backup is performed. The history tables always match the transactions in the current tables and no partial transactions are committed. Even if a transaction is underway at the time of the backup, the database is consistent.

For the most reliable backups for Oracle databases, Oracle recommends running the Oracle database in ARCHIVELOG mode. ARCHIVE mode ensures that your database is protected from both instance and media failure and, because all changes made to the database are saved in a redo log, all database updates are available for recovery rather than just the most recent changes. For SQL Server, Microsoft recommends running the SQL Server database using the full recovery model, which allows a database to be recovered to the point of failure. Online backups are available for Oracle and SQL Server databases running in these modes.

Offline Backups

If needed, you can perform offline backups of the master index database. In this case, you must queue any incoming messages using the JMS IQ Manager and undeploy the master index

application before beginning the backup. Once the backup is complete, restart the database, redeploy the master index application, and then process the messages queued by the JMS IQ Manager.

Restoring the Master Index Database

In the unlikely event that you need to restore the master index database to a previously archived version, you must undeploy the master index application prior to performing the restoration to ensure that the application retrieves the correct sequence numbers from the database once it is restored. Any new transactions that occurred after the archived version was created will be lost, but they can be resent if the JMS IQ Manager is configured to journal all messages.

Archiving Master Index Data

In addition to regular database backups, some of the master index database tables can grow very large. For performance reasons, you might want to archive the information in the sbyn_assumedmatch and the sbyn_audit tables.

Implementing Changes to the Master Index Project

After a master index application has been in production, you might need to make changes to your project. For example, if you add a new external system, you need to add that system to the master index database and you might need to modify the object structure and OTDs as well as update the application files. Changes occur as the needs of your end users evolve and as additional external systems are added. Do not make changes to the system hastily. Handle changes using the same change management process that was originally used to deploy your project. Applying this same process of planning, configuration, testing, migration, monitoring, and reevaluation will help ensure successful updates.

- "Modifying Master Index Configuration Files" on page 28
- "Modifying the Master Index Database" on page 29
- "Modifying Master Index Security" on page 30
- "Modifying the Local ID Format" on page 30

Modifying Master Index Configuration Files

Over time, you might need to make changes to your configuration files, such as adding fields or objects to the object structure, changing queries, or fine-tuning the matching process. Whenever you make a change to a master index configuration file, you must undeploy the master index server project, regenerate the application, and then redeploy the project.

This section provides tips for updating components of the configuration files. In order for any of these changes to take affect, you must regenerate the application and rebuild and redeploy the project.

Updating the Object Structure

If you make any changes to the object structure, keep the following in mind.

- If you want the new fields or objects to appear on the MIDM, make sure to add them to the first section of midm.xml and to any of the page definitions later in the file (this includes search pages).
- If the new fields require normalization, parsing, or phonetic encoding, define the new structures in mefa.xml.
- If a new field will be used for matching, add it to the blocking query used for match processing as well as to the match string in mefa.xml.

Updating Normalization and Standardization Structures

If you define normalization, standardization, or phonetic encoding for fields that are not currently defined in mefa.xml, or if you change existing standardization structures, make sure to do the following.

- Use the appropriate standardization type, domain selector, and field IDs.
- Add the new fields that will store the standardized versions of the original field value to the appropriate objects in object.xml.
- Add new columns to the database to store the standardized field values.

Updating the Match String

If you make changes to the match string, update the database indexes and the blocking query in query.xml accordingly. For example, if you remove a field from the match string, you might also want to remove that field from the blocking query and database indexes. If you add a field to the match string, add the field to the blocking query and to the appropriate database index to maintain performance.

Modifying the Master Index Database

There might be times when you need to modify the master index database. For example, you might need to add or modify a stored procedure or index, or you might need to add new common codes. You must modify the database if you add fields or objects to the object structure in order to reflect the new structure in the database tables. If you make changes to the database, rebuild and redeploy the master index server project to ensure the changes are picked up by the application. The only exception to this is when you add external systems.

Modifying Master Index Security

You can define new users for the database at any time using standard SQL statements to create the type of user you want to define. You can also add new users for the midm.xml through the Sun Java System Application Server. Neither of these procedures require any stoppage of the database or of the master index application and redeployment is not required.

Modifying the Local ID Format

If you need to modify the local ID format for an external system, regenerate the application after you make the changes and then redeploy the project. If you extend the length of a local ID past 20 characters, make sure to increase the length of any database columns containing local IDs. Local ID columns are found in the following tables: sbyn_*parent_object*, sbyn_assumedmatch, sbyn_enterprise, sbyn_systemobject, and sbyn_transaction.