

Matisse[®] 8.0.6

Release Notes

1st Edition

August 2007



Matisse 8.0.6 Release Notes

Copyright ©1992–2007 Matisse Software Inc. All Rights Reserved.

Matisse Software Inc.
930 San Marcos Circle
Mountain View, CA 94043
USA

Printed in USA.

This manual is copyrighted. Under the copyright laws, this manual may not be copied, in whole or in part, without prior written consent of Matisse Software Inc. This manual is provided under the terms of a license between Matisse Software Inc. and the recipient, and its use is subject to the terms of that license.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. and international patents.

TRADEMARKS: Matisse and the Matisse logo are registered trademarks of Matisse Software Inc. All other trademarks belong to their respective owners.

PDF generated 18 July 2007

Contents

1	New Features in Matisse 8.0	5
1.1	Overview	5
1.2	Enterprise Manager Tool	5
	Remote Administration	5
	Server Operation Control	5
	Server Monitoring	6
	Database Monitoring	6
	Object Browser	6
	Backup/Restore	6
1.3	Server Manager Listener	6
1.4	Database Utility Commands	6
	mt_connection	6
	mt_server extendcache	7
	mt_server info	7
	mt_server setlicense	7
	mt_server checklicense	7
1.5	Database Configuration Parameters	8
	NAME	8
	AUTORESTART	8
1.6	Matisse SQL	8
	Returning a Table in a SQL Method	8
1.7	.NET Binding	8
	.NET 2.0 Support	8
	Namespaces Renaming	8
	Class Generator Tool	8
1.8	Java Binding	9
	Java 2 Platform 5.0	9
1.9	XML Tool	10
	mt_xml	10
1.10	Open Source Interfaces and Language Bindings	11
2	Compatibility with Previous Releases	12
2.1	Data Migration	12
	Step 1	12
	Step 2	12
2.2	Client Connections	12
2.3	License Key Format	12
3	Platform-Specific Topics	13
3.1	Linux	13
3.2	MacOS	13

3.3	Solaris	13
3.4	Windows	13
4	Update History	14
	Resolved in Matisse 8.0.6	14
	Resolved in Matisse 8.0.5	14
	Resolved in Matisse 8.0.4	15
	Resolved in Matisse 8.0.3	15
	Resolved in Matisse 8.0.2	16
	Resolved in Matisse 7.0.9	16
	Resolved in Matisse 7.0.5	17
	Resolved in Matisse 7.0.4	17
	Resolved in Matisse 7.0.3	18
	Resolved in Matisse 7.0.2	18
	Resolved in Matisse 6.1.1	18
	Resolved in Matisse 6.1.0	19
5	Documentation	20
	Matisse documents available on the Web	20
	Documents included with Matisse standard installation	20
	Open source bindings	20

1 New Features in Matisse 8.0

1.1 Overview

The Matisse 8.0 release introduces major features in the Matisse product line:

- The *Matisse Enterprise Manager* now provides full remote administration features for distributed Matisse database servers on a local network.
- The *.NET binding* now supports Microsoft's .NET Framework 2.0 and Visual Studio 2005.
- The *Java binding* now supports Sun's Java 2 Platform 5.0.
- Matisse 64-bit is now available on Linux and Windows.
- Support for Solaris 10 on both SPARC and AMD Opteron chips.
- Support for Windows Vista.

The Matisse 8.0 release also includes many improvements in SQL, language bindings, and the Matisse client library.

1.2 Enterprise Manager Tool

The *Matisse Enterprise Manager* has been redesigned to regroup in a single tool: the distributed management of database servers, the management of database schemas, the data import and export in table (relational) and XML formats, as well as various security and administration functions. It includes an Object Viewer to browse and edit object hierarchies stored in a database. It also includes a SQL analyzer tool to help optimize complex queries and produce result-sets in a table format.

Remote Administration

The *Enterprise Manager* now provides full remote administration features for distributed Matisse database servers on a local network. All administration tasks can be executed remotely via the enterprise manager. This includes database start up and shutdown, database backup and restore, server monitoring, database monitoring, database access control and database schema and data manipulation.

Server Operation Control

The Server Operation Control component provides security control for executing local or remote administration operations including start/stop databases, backup/restore and datafile management.

- Server Monitoring** The *Enterprise Manager* monitors registered database servers detecting in real-time when servers, databases and Matisse services become unavailable. It also provides real-time monitoring of CPU activity, memory consumption and disk usage of database servers.
- Database Monitoring** The Database Monitoring component presents real-time detailed information on a selected database. It provides information on the active connections to the database, transactions performed on the database, and data storage activity. This tool replaces the former `mt_monitor` utility.
- Object Browser** The new Object Browser allows users to access object hierarchies stored into a database. Matisse Standard and Advanced Query Builders provide an easy access to objects. You can then navigate through the object result-set and update object properties.
- Backup/Restore** Matisse Database Backup allows users to perform full and incremental parallel backups of databases while the system is online. There is no need to block updates during a backup, as the backup sub-system uses a snapshot of the database at the time of the beginning of the backup operation.
- Matisse Database Restore provides wizards to guide administrators through the restore process. When restoring, you must first start your database for restore, then you can start the restore process. For restoring from a multi-increment backup, you can restore the full backup files and the incremental backup files in any order, either sequentially or in parallel, and then shutdown and restart the database complete the process.

1.3 Server Manager Listener

Matisse Server Manager Listener (SMListener) manages remote operation requests on a local network. The `mt_smlistener` daemon also controls the denial of operations execution on the machine it is running on. The SMListener daemon is responsible for creating new databases, starting and stopping databases as well as managing backups and restore operations. The SMListener utility is also responsible for restarting database servers automatically after a reboot of the machine.

1.4 Database Utility Commands

- mt_connection** The `mt_connection` utility allows you to view connections and kill active connections. A Connection ID field has been added to the connection description. The `--cid` option has been added to the kill command to designate a connection to be killed by its ID. Now connections can be killed individually in a multi-threaded environment with multiple connections.

```
$ mt_connection -d example list
```

2 connection(s):

ID	NODE	USER	PID	CALLS	TRANID	BLOCKED	FUNCTION
7426	JADE	john	20593	1			connections
7424	JADE	mary	20587	26			exec sql

.

mt_server extendcache

The `mt_server extendcache` command allows you to extend the size of the server page cache on a running database.

```
$ mt_server -d example extendcache -s 1024M
Server cache size extended to 1024M
```

mt_server info

The `mt_server info` command, which provides information for a running database, now displays information in a format similar to the Enterprise Manager Tool.

```
$ mt_server -d example info
Database:
  Name: example
  Server version: 8.0.3
  Datafile version: 8.0.3
Status:
  Start date: 05 Jul. 2006 15:33:13
  Server uptime: 52 sec
  Backup date: not available
  Version collect date: not available
Configuration:
  Page size: 8 Kbytes
  Page cache size: 32 Mbytes
  Server failover: disabled
  Datafile extension: automatic
  Access control: disabled
  Version collection: automatic
Current State:
  Transaction manager: enabled
  Current logical time: 2
  Highest collected logical time: 1
  Last backup logical time: 0
```

mt_server setlicense

The `mt_server setlicense` command allows you to set the customer license key on your server. The `setlicense` command replaces the `license` command available in release 7.

mt_server checklicense

The `mt_server checklicense` command allows you to check the customer license key installed on your server. The `--full` option provides a complete description of the installed license.

```
$ mt_server checklicense
Your 31 days license expires in 15 days
```

```
$ mt_server checklicense --full
License Description:
Floating License - Standard Edition - up to 5 users - up to 4 logical CPUs
License options - mirroring disabled, raw partition datafiles disabled, replication disabled
```

License expires in 474 days

1.5 Database Configuration Parameters

- NAME** This parameter defines the name of the database. The maximum number of characters for a database name, which was 12 in prior releases, has been increased to 31 characters.
- AUTORESTART** This parameter establishes the automatic restart of a database when the host machine is rebooted. When this parameter value is set to 1, the Matisse Server Manager Listener restarts the database server automatically upon reboot. When it is set to 0, no action is performed upon reboot.

1.6 Matisse SQL

- Returning a Table in a SQL Method** A new syntax has been added to SQL methods for returning tables. For instance, the following SQL method returns a table listing the presidents in office between startYear and endYear:

```
CREATE STATIC METHOD viewPresidentsBetween(startYear INT, endYear INT)
  RETURNS TABLE(firstName VARCHAR(32), lastName VARCHAR(32), startingYear INT,
    endingYear INT)
  FOR Presidency
  BEGIN
    SELECT p.FirstName, p.LastName, p.IsInChargeOf.StartingYear, p.IsInChargeOf.EndingYear
    FROM Person p
    WHERE p.IsInChargeOf.EndingYear >= startYear
    AND p.IsInChargeOf.StartingYear <= endYear;
  END;
```

1.7 .NET Binding

- .NET 2.0 Support** Matisse .NET binding has been upgraded to support Microsoft's new .NET Framework 2.0 and Visual Studio 2005. In Release 8, the support for .NET Framework 1.1 has been discontinued.
- Namespaces Renaming** To align Matisse .NET binding to Microsoft .NET framework naming conventions, com.matisse namespace has been renamed Matisse and com.matisse.reflect namespace has been renamed Matisse.Reflect.
- Class Generator Tool** The Matisse .NET Object Manager utility mt_dnom generates C# or VB.NET source codes. This utility generates source codes for calling SQL methods in the database server without using a SQL statement, which seamlessly extends object-

oriented programming to the database server. 'ADO Data Classes' can be also generated in addition to the regular stub class generation by the `mt_dnom` utility. The 'ADO Data Classes' are useful when you need to copy the Matisse property values to your application classes that are independent from the persistence layer, for example serializing objects for transfer over the network. The `mt_dnom` utility now supports user authentication.

```
$ mt_dnom
```

```
Matisse .NET Object Manager x32 Version 8.0.3 (32-bit Edition) - Sep 21 2006.
```

```
(c) Copyright 1992-2006 Matisse Software Inc. All rights reserved.
```

Usage:

Generate Stubs:

```
mt_dnom -d [user:]database[@host[:port]] [-p] stubgen [-lang C# | VB] [-n <namespace>] [-adc <namespace>] [-no]psm]
```

```
-d [user:]database[@host] : the database to be accessed.
```

```
-p : allows the user to authenticate with a username/password.
- if the '-p' option is used, it will be assumed that the current system user is known from the database, but a password will be asked.
- if the '-p' option is not used, Matisse <user> is used for user name, and a password will be asked.
- if the user is not defined and the '-p' option is not used, it will be assumed that the current system user is known from the database, and does not need password.
```

```
stubgen [-lang C# | VB] [-n <namespace>] [-adc <namespace>] [-no]psm]
```

```
-lang : generate C# or VB files from the database schema. The default is C#
```

```
-n <namespace> : define the generated classes in the specified namespace
```

```
[-no]psm : generate .NET methods mapping Persistent SQL methods. The default is -psm
```

```
-adc <namespace> : generate ADO Data Classes in addition to stub classes
```

The Matisse .NET Object Manager utility `mt_dnom` utility replaces the former `mt_stbgen` utility.

1.8 Java Binding

Java 2 Platform 5.0 Matisse Java binding has been upgraded to support the new Sun Java 2 Platform 5.0.

1.9 XML Tool

mt_xml

The Matisse XML import/export utility has been updated to provide command line options similar to the other Matisse utilities.

```
$ mt_xml
MATISSE XML Manager x32 Version 8.0.3 (32-bit Edition) - Sep 21 2006.
(c) Copyright 1992-2006 Matisse Software Inc. All rights reserved.
```

Usage:

Import:

```
mt_xml [-v] -d [<user>:]<database>[@<host>[:<port>]] [-p]
import {-f <xml_file> | -in} [-update] [-commit <n>]
```

Export:

```
mt_xml [-v] -d [<user>:]<database>[@<host>[:<port>]] [-p]
export {-f <xml_file> | -out} [-foid] {-full | -sql <stmt> | -oid <oid>...}
```

-d [user:]database[@host]: Specifies the database to be accessed.

-p : Allows the user to authenticate with a specific username/password.
- if the '-p' option is used, it will be assumed that the current system user is known from the database, but a password will be asked.
- if the '-p' option is not used, Matisse <user> is used for user name, and a password will be asked.
- if the user is not defined and the '-p' option is not used, it will be assumed that the current system user is known from the database, and does not need password.

-v : Reports the number of objects imported/exported.

import -f <xml_file> : Import the XML data into the database.
The export option allows to generate this file containing XML data extracted from the database. Arguments appearing within {} indicate that one of the arguments must be specified.

import -in : Reads the XML data from standard input and import it into the database.

-update : When specified with MtPrimaryKey attribute, values of existing objects are updated.

-commit <n> : Commits transaction for every <n> objects parsed.

export -f <xml_file> : Generate the XML file containing XML data extracted from the database by specifying either a SQL statement <stmt> or by <oid>s.

Arguments appearing within {} indicate that one of the arguments must be specified.

- `export -out` : Writes XML data to the standard output.
- `-full` : Exports all non-schema data into a single XML file.
- `-sql <stmt>` : Specifies objects to be exported using a SQL statement <stmt>.
- `-oid <oid>` : Specifies OIDs of objects exported. Both decimal and hexadecimal oid are accepted.
- `-foid` : Exports data in a format with OIDs in the xml tags to enable Primary Key recovery. (The `-full` option always exports in this format)

1.10 Open Source Interfaces and Language Bindings

The following bindings are currently available in open source and have been updated for release 8:

- Eiffel binding 8
- Perl binding 8
- PHP binding 8
- Python binding 8
- Smalltalk binding 8

These bindings are all provided as open source. Developers are encouraged to contribute to these bindings under Matisse Interface Public License (MIPL), which is a variant of the Mozilla Public License.

You may check the download page for the current status of the open source bindings:

<http://www.matisse.com/developers/downloads/>

2 Compatibility with Previous Releases

2.1 Data Migration

Matisse Server 8.0 comes with several changes in the data format. You must use the `mt_xml` tool for converting an existing database (7.x or prior) into the 8.0 format.

Step 1

Before installing 8.0.x, save your schema in ODL and your data in XML format:

```
mt_sdl -d <dbname> export -odl schema.odl
```

```
mt_xml -d <dbname> export -f data.xml -full
```

Prior to 8.0.4, use the command below:

```
mt_xml -d <dbname> -xml data.xml -full
```

You may check the [Matisse® XML Programming Guide](#) for more options with exporting in XML format.

Step 2

You may now install Matisse 8.0.x on your machine and then restore the schema and the data as follows:

```
mt_sdl -d <dbname> import -odl schema.odl
```

```
mt_xml -d <dbname> import -f data.xml
```

2.2 Client Connections

Only 8.0.4 clients may be used with 8.0.4 servers.

The clients for earlier releases of Matisse are incompatible with the 8.0.4 server. Consequently, you must upgrade any older clients to 8.0.4 before attempting to access an 8.0.4 server.

2.3 License Key Format

The customer license key format has changed in release 8. Matisse 8 does not recognize license keys issued for prior releases. Upon installation of Matisse 8, a license key valid for 30 days is automatically issued.

3 Platform-Specific Topics

3.1 Linux

The following Linux distributions on x86 (32-bit) and AMD64 and EM64T (64-bit) chips families are supported:

- Red Hat Enterprise Linux 4
- Fedora Core 3 / 4 / 5 / 6
- SUSE Linux Enterprise Server 9
- SUSE 9.2 / 9.3
- CentOS 4.1 / 4.2 / 4.3 / 4.4

Any other Linux distributions, where Matisse 8 has not been tested, require Linux kernel 2.6.9 on systems based on x86 (32-bit) or x86_64 (64-bit) chips families.

3.2 MacOS

The MacOS X version for Intel of Matisse is available upon request.

3.3 Solaris

Support for Solaris 10 on SPARC with 32-bit kernel and 64-bit kernel.

Support for Solaris 10 on x86 (32-bit) and AMD64 (64-bit) chips families.

Installation packages for Solaris SPARC with 64-bit kernel are available upon request.

3.4 Windows

Support for Windows (including Vista) on systems based on x86 (32-bit) and x86_64 (64-bit) chips families.

The Matisse x64 version for Windows 64-bit is available upon request.

4 Update History

This section contains the list of bug fixes and minor feature changes between releases. You may refer to it before upgrading to see if the new release resolves a known problem or adds a needed feature.

Resolved in Matisse 8.0.6

- Support for Windows Vista.
- Matisse Enterprise Manager auto-refresh option does not update the database schema tree to reflect the schema changes made by an external program or utility (i.e. mt_sdl).
- In Matisse Enterprise Manager, the Object Browser feature presents a Statistics panel to display the object count for each class in the database.
- “Add mirror” button and menu have been added to the Matisse Enterprise Manager Datafiles feature to ease the management of mirrored datafiles.
- The Matisse Enterprise Manager Datafiles feature now displays a Datafile Status column.
- On Windows, ‘MATISSE-E-SQLSTACKOVF, MATISSE SQL stack over flow’ could be returned when executing a SELECT statement.
- Establishing (one-2-many or many-to-many) relationships with mt_dts on classes with a few millions of objects could require to extend the data-files to a large size to complete in a reasonable amount of time.
- Matisse Data Transformation Services (mt_dts) did not allow import of attribute list types listed as one element per line.
- Matisse Data Transformation Services did not allow primary key-based updates. A new import option tag has been added: AllowUpdates.

Import Tag: <allowUpdates>YES</allowUpdates>

- Matisse Data Transformation Services (mt_dts) did not allow to import/export large binary data (MT_BYTES) and large text data (MT_TEXT) into files. By default, these 2 types are still imported/exported as field values. But using the ColumnFromFile and ColumnToFile option tags allow to manage the values into files.

Import Tag:

<ColumnFromFile Directory="/import/documents">Document</ColumnFromFile>

Export Tag

<ColumnToFile Directory="/export/documents" FilenameFormat="{DocIdent}.txt">Document</ColumnToFile>

Resolved in Matisse 8.0.5

- Matisse Data Transformation Services (mt_dts) did not return a warning when a value is missing on the last row of the file and that the row does not include a newline character.

- On Windows, the environment system variables MATISSE_CFG and MATISSE_LOG defining alternate directories for the Matisse config and log files were ignored.
- Matisse Enterprise Manager could not display UTF16 String attribute values in the Object Browser.
- 'MATISSE-E-INTERNALERROR, RS PropVal opts' not cached error could be returned at commit when updating multiple relationships of an object after a class update including removing existing relationships and adding new ones.
- In some cases, an object could have a relationship pointing a non-existent object preventing any navigation or modification of the relationship of this object.

Resolved in Matisse 8.0.4

- On Windows the maximum size for a backup file slice was limited to 2Gbytes. This limit has been removed.
- On Windows, Matisse Enterprise Manager and Matisse Server Manager Listener did not always start properly when multiple versions of the Java Virtual Machine are installed on the machine and listed in the Path.
- Matisse Enterprise Manager could not export data from class 'Movie' in the 'Media' sample database
- In Matisse Enterprise Manager, when adding a mirror datafile with a file size not identical to the primary datafile, the invalid capacity error message was misleading.
- In Matisse Enterprise Manager Object Browser, the paging viewer did not always appear when a SQL query returned a large collection of objects.
- In Matisse Enterprise Manager Object Browser, an Advanced tab has been added to make both standard and advanced query builders more independent of each other.
- Matisse Data Transformation Services (mt_dts) did not support importing/exporting multimedia data into files. 2 XML options tags have been added: ColumnFromFile and ColumnToFile

Import Tag:

```
<ColumnFromFile Directory="/import/images">Photo</ColumnFromFile>
```

Export Tag

```
<ColumnToFile Directory="/export/images"      FilenameFormat="img_{LastName}.jpg">Photo</ColumnToFile>
```

- Matisse Data Transformation Services (mt_dts) did not allow to link classes with primary/foreign keys composed of multiple columns.
- SQL DDL script generated with mt_sdl did not add double-quote characters (") around every attribute and class names which could collide with SQL reserved keywords (i.e. Group should be "Group")
- in some cases, a SQL method returning a large object collection could include a null object.
- MATISSE-E-OBJECTNOTFOUND, 0x0 not found' error could be returned at commit when updating an object with multiple ordered relationships with a very large number of successors.

Resolved in Matisse 8.0.3

- In the Enterprise Manager tool, icons were not displayed in the Index Name and Criteria[1-4] columns of the Indexes table as well as in the Entry Point Name and Attribute Name columns of the Entry Point Dictionaries table when presenting the properties of a Class.
- In some cases, accessing a MT_STRING_LIST value using Matisse C++ Binding on a 64-bit platform could lead to a memory corruption.
- All tools and utility commands print out a message which distinguishes clearly the 64-bit edition from the 32-bit edition.

```
$ mt_emgr -V
```

```
MATISSE Enterprise Manager x32 Version 8.0.3 (32-bit Edition) - Sep 21 2006.  
(c) Copyright 1992-2006 Matisse Software Inc. All rights reserved.
```

```
$ mt_emgr -V
```

```
MATISSE Enterprise Manager x64 Version 8.0.3 (64-bit Edition) - Sep 21 2006.  
(c) Copyright 1992-2006 Matisse Software Inc. All rights reserved.
```

- Added --full option to the mt_server checklicense command to provide a complete description of the installed license.

Resolved in Matisse 8.0.2

- MATISSE-E-SYSTEMERROR, System error 0x846817a could be returned when opening a connection to a remote server protected by a firewall.
- Some nodelocked license key installed on Windows machines with multiple active Ethernet cards can prevent database servers to restart after a reboot of the host machine.

Resolved in Matisse 7.0.9

- In some concurrent access cases, the MATISSE-E-WRITEWAITTIME error is returned before completion of the delay set by the user.
- MATISSE-E-SYSTEMERROR could be returned at commit time when large number of objects with variable size key indexes were deleted in one transaction.
- In SQL, the default memory quota has been increased from 10M to 500M eliminating the MATISSE-E-SQLSTACKOVF error when manipulating a few millions of objects in a single transaction.
- Added a new SQL statement: "SET MEMORY_QUOTA x" where x represents the maximum size of the server-side SQL workspace for a given connection. It is expressed in Mega Bytes. The default value is set to 500M. The minimum value is 50M. Note that setting a large value may cause a performance degradation if your server has insufficient memory.
- In SQL PSM, Stored Methods returning a list of objects could in some cases re-order by OID the object list instead of keeping the order defined by the user.
- Added --oldpasswd option to the mt_user passwd command so that it can be used in non-interactive mode.

- In Java the method `getBytes` defined on `MtObject` did not return a correct value. Generated methods for persistent classes with attributes of type `BYTES` are also affected by this issue.

Resolved in Matisse 7.0.8

- Restoring backups could fail with one of the 2 following errors:
GOM-F-NOENVROOTOBJ, Database root object not found – must init or restore
GOM-E-OBJNOTFOUND, Object <OID> not found

Resolved in Matisse 7.0.7

- Access to large relationships could fail with an internal Matisse Error (`MATISSE-E-INTERNALERROR`, Wrong data tag read).

Resolved in Matisse 7.0.6

- The .NET stub generator (`mt_stbgen`) does not generate correct data class definition when the schema class does not include any attribute or relationship.
- The `IDataRecord.GetBytes` method of the ADO.NET provider does not work properly when retrieving `byte[]` values.
- JDBC does not return blob type values.
- Stereotypes in Rational Rose UML are not handled properly.
- More scalar types have been supported by the Matisse Rose Link.
- The SQL `DROP CLASS` statement fails when the class contains self-referencing relationship
- The SQL compilation fails when a query statement has complex aggregation expression with relationship navigation and built-in function.
- OID comparison in SQL query has been enhanced.

Resolved in Matisse 7.0.5

- Within the Enterprise Manager's query analyzer, SQL statements containing `'{'` in a literal constant string do not execute because of the JDBC escape processing.
- Memory leaks in the Matisse client when executing SQL statements in some cases.
- SQL `SELECT INTO` statement may not work properly when used repeatedly within a stored method.
- The `mt_odl` utility does not work for a global relationship defined between classes.
- The `mt_xml` utility does not work properly when importing an XML document that misses values for the XML attribute `prealloc` within processing instructions.
- Establishment of relationships in the Data Transformation Services does not work properly for some attribute types.
- Concurrent request of establishing database connections could fail when the number of requests is large because of operating system's limitation.

- Eiffel binding: {MT_LINKED_LIST}.wipe_out raises an error when the list is empty

Resolved in Matisse 7.0.4

- Missing MOD(Dividend, Divisor) built-in to calculate modulo in SQL.
- On Windows, the Enterprise Manager doesn't start when the current user account is different from the user account at installation.
- Enterprise Manager "Import Data..." command doesn't report errors properly.
- Enterprise Manager "Establish Relationships" command doesn't work when PK/FK are defined on fixed size data types.
- Some SQL queries do not display the result correctly when executed from the Enterprise Manager.
- SQL built-in function CONCAT does not return a correct result when a parameter and the result variable are identical and the function is executed repeatedly in SQL Methods.
- The mt_stbgen utility for the .NET binding does not return error messages properly.

Resolved in Matisse 7.0.3

- The right click menu on the monitor window in the Enterprise Manager does not freeze the refresh, thus making it difficult to select a single line, for instance to abort a transaction.
- Several templates in the Enterprise Manager SQL Analyzer window are inaccurate or contain syntax errors.
- The mt_stbgen tool generates extra closing curly braces in some cases, when re-generating the stub classes.
- ODL export is including the Meta-Schema, both from Enterprise Manager and from the mt_sdl tool.
- Delete statements in stored methods in some cases ignores proper relationships of sub-classes involved in the deletion, thus leaving unbalanced inverse relationships.
- The SELECTION expression cannot be used in the FROM clause of a SELECT statement. For instance: "SELECT * FROM SELECTION(Sel1 UNION Sel2) AS Class WHERE ...".

Resolved in Matisse 7.0.2

- 'mt_version -n' removes all versions instead of only the one specified.
- Removing a Datafile may block and not be able to complete in some cases.
- Java binding date and timestamp cannot exceed year 2038.
- Empty strings not indexed in entry-point dictionaries.
- Spaces in Datafile path not supported.
- Concurrent update and delete of objects causes deadlock even in 'access for update' mode.

- The .NET code generation utility `mt_stbgen` may not generate correct code for SQL methods when there are overridden methods.
- SQL `DROP CLASS` statement does not delete related indexes and entry point dictionaries.
- SQL built-in function `COUNT` used with a relationship, which returns the number of successor objects in the relationship, returns `NULL` instead of `0` when the relationship is empty.
- Java binding may not properly release internal resources after disconnecting from a database.

Resolved in Matisse 6.1.1

- Date and timestamp in the Java binding do not support full range of available values.
- SQL `SELECT REF()` may not work in some specific environment.

Resolved in Matisse 6.1.0

- XML import truncates large attributes.
- ADO.NET interface `IDbConnection.ConnectionTimeout` returns `0`. It returns `1` in the new release.
- The .NET binding may not free external resources for database connection when garbage collector calls the `Dispose` method.
- Cannot obtain the class in the `FROM` clause from `MtSQLGetStmtInfo`.
- SQL syntax doesn't allow `LIST (INT)` instead of `LIST (INTEGER)`.
- Schema import/export as `odl` in editor doesn't work with empty spaces in path.
- Database deadlocks during SQL execution may cause memory leaks on server side.
- `MtSQLSetCurrent` does not position on the relevant row in some cases, this also affects ODBC access.
- Some datatypes have errors when updating with ODBC `SQLSetPos`.
- User license key checks does not work properly for some hardware configurations.
- Large transactions with entry-point dictionary maintenance may cause an RPC error.
- With the Eiffel binding, putting an object in a specific place in `MT_LINKED_LIST` causes an error on post condition.
- When deleting objects with the Eiffel binding, some class invariants do not work correctly.
- The Eiffel binding may not free external resources correctly when garbage collector calls the `Dispose` method.

5 Documentation

Matisse documents available on the Web

The following documents are available at <http://www.matisse.com/developers/documentation>:

- Installation guides for Linux, MS Windows, and Solaris
- *Getting Started with Matisse*
- *Matisse SQL Programmer's Guide* (includes user's guide for mt_sql)
- *Matisse .NET Programmer's Guide* (and example applications)
- *Matisse Java Programmer's Guide* (and example applications)
- *Matisse C++ Programmer's Guide* (and example applications)
- *Matisse C API Reference*
- *Matisse ODL Programmer's Guide*
- *Matisse Rose Link User's Guide*
- *Matisse Server Administration Guide*
- *Matisse XML Programming Guide* (includes user's guide for mt_xml)
- *Matisse Editor User Guide for MS Windows* (user's guide for mt_editor)
- *Matisse Editor User Guide for X/Motif* (user's guide for mt_editor)

Documents included with Matisse standard installation

- Guide to Matisse documentation and other resources: `readme.html`
- *Matisse .NET Binding API Reference*: `docs/NET/Solution_MatisseNet.HTM`
- *Matisse Java Binding API Reference*: `docs/java/api/index.html`
- *Matisse C++ Binding API Reference*: `docs/cxx/api/index.html`

Open source bindings

- *Matisse Eiffel Programmer's Guide*
- *Matisse Perl Programmer's Guide*
- *Matisse PHP Extension Reference*
- *Matisse Python Interface Reference*

The Matisse Smalltalk documentation is included in the Matisse Smalltalk binding package.