

erwin Data Modeler 2021R1

Meta Integration Model Bridge Known Limitations

This document provides information about known limitations of the erwin Metadata Management (EMM) Solution, erwin Web Portal (EWP).

EMM Metadata Harvesting with with Meta Integration® Model Bridge (MIMB) has the following known limitations:

Overview

The erwin Metadata Management (EMM) solution, erwin Web Portal (EWP), provides all essential features for erwin data model publishing, analysis, and reviews:

- Metadata harvesting (integration) from erwin Data Modeler Standard Edition
- *Metadata harvesting (integration) from erwin Data Modeler Workgroup Edition (synchronizes with the erwin Mart repository).*
- Data Model Diagram Visualizer and Navigator
- Metadata (single) Configuration and Stitching
- Metadata Browsing, Search and Reporting
- Metadata Collaboration (external URL, tagging, comments and review)
- Data Flow Lineage & Impact Analysis
- Metadata Explorer (simplified metadata user interface for business users)
- *Repository Security with User and Role Management (integration with Enterprise LDAP or SSO)*
- *Repository Customization by adding user defined properties / attributes)*
- *Repository Storage on PostgreSQL (bundled on Windows), Oracle or SQL Server*

Note: The features mentioned in italic have been added from the previous CA ERwin Web Portal Enterprise Edition.

Copyright Notice

erwin Web Portal (EWP) is licensed under the following copyright:

Copyright © 2021 erwin, Inc. All rights reserved.
Copyright 2021 Quest Software Inc. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
erwin® is a registered trademark of erwin, Inc.

erwin Web Portal is based (re-branded OEM) on the following Meta Integration products:

- Meta Integration® Metadata Management (MIMM)
- Meta Integration® Model Bridge (MIMB)
- Meta Integration® Repository (MIR)

These products are licensed under the following Meta Integration copyright:

Copyright © Meta Integration Technology, Inc. 1997-2021.
All Rights Reserved.
Meta Integration® is a registered trademark of Meta Integration Technology, Inc.
Other product and company names (or logos) mentioned herein may be the trademarks of their respective owners.
<http://www.metaintegration.com>

MIMB Known Limitations

The erwin Metadata Management (EMM) solution, erwin Web Portal (EWP), provides all essential features for erwin data model publishing, analysis, and reviews:

1. Consistency Check of the Source Model
2. Metadata Import/Export is limited to the "Bridge Specifications"
3. Potential Loss of Metadata from Tool A to Tool B
4. Limitations on Import/Export Round-Trip
5. Limitations on Export Bridge's Merge/Update Capabilities
6. API based Bridges
7. Multi-file based Bridges
8. Graphical Layout is carried only through some Bridges
9. SQL Parsing of Transformation Expressions in Import Bridges
10. Conversions of Transformation Expressions in Export Bridges
11. Internationalization Aspects
12. Multi-Model Import Bridges for Incremental Metadata Harvesting
13. Linux vs. Windows Only Availability
14. Third-party software (drivers) internet download

Consistency Check of the Source Model

BACKGROUND PURPOSE:

Data Modeling, Data Integration (ETL/DI), Business Intelligence (BI) and other design tools supported by MIMB may not always save models which are valid. Indeed, some of these models may still be under development (not yet completed), or the model validation may be a separate step (design tool menu entry, separate tool, or part of the model publication). Therefore, MIMB import bridges may be dealing with semantically incorrect or incomplete models which the MIR repository can save, but they may break the export bridges. Having a centralized model validation (or consistency check) between the import and export phases allows to continuously improve the consistency check algorithms, and avoids having each export bridge retest for the same model validity issues.

INPUT PARAMETERS:

The user has the choice of 3 levels of consistency check:

- **"No Consistency Check"**: If the user purposely decides to turn off all consistency check (e.g. when the tool is not available to fix the source model), then the imported model may break the export bridge, or produce an invalid model that may break the target tool. MITI assumes no responsibility when consistency check is turned off.
- **"Basic Consistency Check"**: This level corresponds all minimal consistency checks required to export, including (but not limited to) the detection of disconnected relationships, foreign keys not connected to any primary or alternate key, etc.
 - Note 1: the source model may pass the model validation provided by the original tool, but still may fail the basic consistency check which may include more substantial testing required by the target tools.
 - Note 2: the basic consistency check may be more rigorous than needed by the export bridge or target technology/tool that will be used. In such case, some of the warnings or errors can be ignored.
 - Note 3: the basic consistency may not be complete enough to catch all the model validation tests of the target tools. Therefore, the user should still run any model validation or consistency check available in the target technology/tool.
- **"Detailed Consistency Check"**: This level includes a more verbose version of the above basic consistency checks, plus more advanced semantic checks specific to the technology of the source model: RDBMS (e.g. attributes of a PK cannot have a null value), ETL/DI metamodels, or ROLAP/MOLAP BI metamodels (e.g. a hierarchy must have a root). This level is sometimes used as a substitute to the lack of model validation capabilities in the original tool, but is not required to use the MIMB export bridges.

OUTPUT RESULTS:

Consistency checks will report (in the log) "WARNINGS" on the source model, or "ERRORS" that are fatal and will prevent the export to any tool. The solution is to fix the original model, and then import it again.

Metadata Import/Export is limited to the Bridge Specifications

The metadata mapping and transformations implemented by Meta Integration Model Bridge (MIMB) are limited to the published model mapping specifications for each integrated tool. Such specifications are part of the MIMB online help, and are also published on the web at <http://www.metaintegration.net/Products/MIMB/SupportedTools.html> (select/click on a bridge to get the full specifications including all the bridge parameters/options and detailed mapping specifications).

The metadata mapping specifications are intended to define the scope of metadata captured (by an import bridge) or populated (by an export bridge) in the 3rd party tool or format (tool and metamodel naming). Therefore, it can only be used to analyze the scope of the metadata covered by the bridge algorithms. However, it is not intended to be a complete documentation of all the mapping algorithms involved in migrating metadata from any tool to any other tool, as the algorithms includes expression parsing, expression conversions and other complex operations which may not be expressed as a mapping specification.

Potential Loss of Metadata from Tool A to Tool B

There are 7 points of potential metadata loss from Tool A to Tool B:

1. Tool A's Methodology/Metamodel Support & Implementation,
 2. Tool A's Export Capabilities,
 3. MIMB's Import Bridge for Tool A,
 4. MIR's Methodology/Metamodel Support & Implementation,
 5. MIMB's Export Bridge for Tool B,
 6. Tool B's Import Capabilities,
 7. Tool B's Methodology/Metamodel Support & Implementation.
- As far as (1) and (7) are concerned, tools may have different implementations of the same methodology (IDEF1X or UML). For example some concepts may not have been implemented yet, or some proprietary extensions may have been done. We usually deliver a demo or generic coverage model for each tool in the "Samples" directory of the MIMB installation.
 - As far as (2) and (6) are concerned, it is always a good idea to export a model from the tool and immediately re-import it to check if all modeling concepts are properly exported/imported. The Import/Export capabilities (file format or API) are not always the best-tested features of a tool. It makes business sense for design tool suppliers to put more priorities on functionalities, robustness, and performance, in order to keep a competitive advantage.

- As far as (4) is concerned, MIMB uses a non-persistent version of the Meta Integration Repository (MIR) which implements and integrates IDEF1X data modeling, OMG UML object modeling, and OMG CWM warehouse modeling (see <http://www.metaintegration.net/Products/MIRSDK/> for more details). There are the current limitations:
 - MIR does not implement all IDEF, OMG UML & CWM standards. MITI has been focusing on data and basic object modeling for its metadata and data movement solutions. The support for activity/process modeling is not implemented (due to the lack of agreements and possible mapping between methodologies in that area). Therefore:
 - MIR implements IDEF1X data modeling, but does not implement the other IDEF standards such as IDEF0 activity modeling.
 - MIR implements UML Class Diagrams, but does not implement the other UML diagrams such as Use-Case, State-Transition, or Interaction diagrams.
 - MIR implements CWM Relational, OLAP and Transformation packages, but does not implement some the CWM metamodels such as Warehouse Process, or Warehouse Operation packages
 - MIR implements conceptual, logical, and physical modeling concepts in an integrated manner. For example, the primary/foreign key of an IDEF1X physical data model is transferred as qualifiers of a UML logical object model. However MIR currently does not support multiple physical models for the same logical model.
 - MIR defines some standard data types based on ODBC (in order to properly provide data movement solutions). Some tools may provide extra proprietary data types that cannot be properly mapped to MIR data types.
- As far as (3) and (5) are concerned, the metadata mapping of both import & export bridges of each tool can be found in the MIMB documentation: on-line help or through the MITI web site at <http://www.metaintegration.net/Products/MIMB/SupportedTools.html> (select/click on a bridge to get the full specifications including all the bridge parameters/options and detailed mapping specifications).
 - The Database Data Type support is limited by the published Data Type Mapping Specifications at <http://www.metaintegration.net/Products/MIMB/MIRDataTypeMapping.html>. Such limitations are based on the supported database or data store technologies, their supported versions, their supported data types, and finally on the actual data type mapping implemented by the import or export bridge. All Data Type Mappings are saved in XML files (within the configuration directory) which are dynamically read by all import and export bridges.
 - The Database specific (proprietary) physical properties are not fully converted, as the information is not always portable across tools' target databases.

MITI provides the best possible metadata mapping across methodologies and tools. However, some modeling concepts may not be available in Tool A, MIR, or Tool B. In such a case, the import & export bridges try to use the descriptions or notes to carry such modeling concepts all the way from the source to the target tool without any loss.

Limitations on Import/Export Round-Trip

There are an additional considerations when performing what is often referred to as a "round-trip", or importing from tool A, exporting to tool B, editing the model (or not) in tool B, re-import this model in from tool B, and exporting back to tool A. In this case, the 7 points of potential metadata loss identified above are compounded so that there are now 7 additional points of loss:

- They involve methodology changes (e.g., UML back to IDEF1X)
- They may contain "Many to one" or "many to many" type relationships (e.g., data type information and data base definitions many be much narrower in one tool than another)
- They are based upon different levels of information in the import process as opposed to the export process (or vice-versa) used to determine the result (e.g., expressions)
- In addition, the tool's own transformations and metadata coverage (even medium for exchange) will likely be different on import versus export (e.g., import may be using an API with information about the entire tool repository available for the bridge, while export is via an XML file which then must be parsed by the receiving tool itself). Again please refer to the web site for this information <http://www.metaintegration.net/Products/MIMB/SupportedTools.html> (select/click on a bridge to get the full specifications including all the bridge parameters/options and detailed mapping specifications).

The fidelity of some simple types of metadata may be well maintained, where there is a direct, one-to-one, translation from tool A --> MIR's methodology --> tool B and then back again. Examples include tools with the same methodology and metadata coverage (e.g., a data modeling and design tool to another data modeling or design tool), or simple elements like the name, description, layout, etc. This process is referred to as "round-tip migration", and may work successfully given these restrictions.

However, it is never recommended that a combination of import-> export-> edit-> export-> import (i.e., "round-trip engineering") be expected to retain fidelity when there is a migration from tool A to tool B (and vice-versa) of methodology, metadata coverage or any of the known limitations identified above. What round-trip engineering implies is a forward-engineering and then reverse engineering process, which has never been a reality in systems engineering. One should not expect the forward engineering of simple to complex transformations (e.g., forward engineering from a spreadsheet to an ETL tool and then reverse engineering from the ETL tool back to a spreadsheet) would be reversible, or even bear much resemblance to the original.

The proper process is to treat the forward-engineering step as "irreversible", treat the reverse engineering step as a separate process, and not expect to effectively compare the original will the "round-trip" result, nor re-use the result for further engineering of the metadata.

Some examples:

- The process of importing from Informatica PowerCenter and then exporting to Microsoft Excel is in fact a "Mapping Requirement Reporting" exercise consisting of reverse engineering a very complex and large graph of transformations steps into a summary of direct mapping lineage in Excel
- The process of importing an ETL design in Microsoft Excel and exporting it to Informatica PowerCenter is in fact a "Mapping Requirement Forward Engineering" exercise to allow business users to automatically prepare the work for the technical designers of the Informatica design tools

Although each of the above steps can use the same Excel format (MetaMap), the round-trip re-engineering is not recommended and is likely to produce results which are not useful for all of the reasons identified above.

Furthermore, like most destination tools, Informatica PowerCenter, as an example, is unable to consume the exported metadata correctly, i.e., it is unable to compare/merge/integrate different versions of complex

transformations. (Please see the section on Limitations in the Receiving Environment for Merge/Update of Metadata.)

Limitations on Export Bridges Merge/Update Capabilities

The export bridges generally depend upon the tool receiving the metadata to integrate the metadata correctly, i.e., provide compare, merge and integrate capabilities. Such capabilities are:

- Frequently available on data modeling and model design tools (e.g. Erwin Data Modeler)
- Available in some ETL tools to source/target data stores, but not for data transformations (e.g. Informatica PowerCenter)
- Rarely available on Business Intelligence design tools (e.g. SAP BusinessObjects Designer)

In order to accommodate these limitations in the destination tool, the following MIMB export bridges offer "limited" metadata update capabilities (Please see the bridge parameter tooltips for more details):

- Data Integration (DI) tools (for ETL Transformations), such as:
 - Microsoft SQL Server Integration Services (SSIS)
 - Oracle Warehouse Builder (OWB)
- Business Intelligence (BI) design tools, such as:
 - Microsoft SQL Server Analysis Services (SSAS)
 - SAP BusinessObjects Designer

API based Bridges

Some MIMB bridges rely on the tool's API to import/export metadata, such as:

- erwin Data Modeler's binary native file formats (.er1 or .erwin) import requires access to the Erwin Data Modeler COM API based SDK.
- IBM InfoSphere Data Architect (IDA) import requires access to the IDA Eclipse based SDK.
- Teradata Database import requires access to the Teradata JDBC drivers.

Such MIMB bridges require these tools to be installed with their API properly setup the machine (PC) where the MIMB bridge is executed, which sometimes mean that the tool's client or desktop version needs to be installed with a valid license (e.g. Erwin Data Modeler) The bridges requiring the tool's SDK/API are annotated with a tool set icon on the published supported tools at: <http://www.metaintegration.net/Products/MIMB/SupportedTools.html>.

Multi-file based Bridges

When multiple files are involved, all must be accessible to the MIMB bridge.

Some MIMB bridges have parameters of type "Directory" instead of just "File", and therefore require access to a directory of multiple files, for example:

- CA Gen (*.ief)
- IBM InfoSphere Data Architect (*.dbm, *.ldm, *.ddm)
- QlikTech QlikView (*.qvw, *.qvd)

Some MIMB bridges have parameters of type "File" which support the notion of "include" files, and therefore require access to all the specified include file paths, for example:

- IBM Rational Rose (MDL)
- XML Schema (XSD) and Document Type Definition (DTD)

Graphical Layout is carried only through some Bridges

Some MIMB bridges convert the graphical information between tools, including conversion of the model layout between various notations like IDEF1X data modeling, and UML object modeling. The following MIMB bridges carry graphical information:

- erwin Data Modeler
- CA COOL:BusinessTeam (Sterling GroundWorks) for some versions
- CA COOL:DBA (Sterling Terrain) for some versions
- IDERA ER/Studio
- IBM Infosphere Data Architect
- IBM Rational Software Architect
- IBM Rational Rose Object and Data Modeler
- NoMagic MagicDraw (via OMG UML 2 XMI)
- Oracle Data Modeler
- SAP BusinessObjects Designer
- SAP (Sybase) PowerDesigner CDM and PDM
- IBM (Telelogic / Popkin) System Architect

However, some MIMB bridges do not transfer the graphical information of the model layout. The primary reason for this limitation is that the import / export capabilities of most tools do not provide graphical information. In other words, their published file formats and/or Application Programming Interface (API) cover the semantic (each of the modeling concepts), but not the graphical information (i.e. the concept's associated shape sizes and positions). Furthermore, when available, such graphical information is not easily reusable in the target tool. This problem is also true for tools sharing the same methodology (e.g. a tool may allow a graphical layout that is not graphically implementable in another equivalent tool). This problem is accentuated when crossing methodology boundaries (e.g. IDEF1X to UML).

- MIMB focuses in moving the full model semantic between tools. The graphical information is then reproduced by the auto-size and auto-layout capabilities of the target tool. For example, erwin Data Modeler may automatically perform an auto-layout when importing an ERX or XML file, and IBM Rational Rose provides a "Layout Diagram" and "Autosize All" features in its "Tools" menu. Some OLAP/BI tools offer more sophisticated auto layout features such as IBM Cognos Framework Manager (FM) with an option to auto layout as a star schema.

SQL Parsing of Transformation Expressions in Import Bridges

Most Database, Data Integration (DI), and Business Intelligence (BI) import bridges depend on a common "SQL Transformation Expression Parsing" library to extract the data flow lineage out of the many transformation expressions used by these technologies. In fact, each tool and technology has its expression syntax. DI and BI tools often defined their transformation expression languages with a syntax close to programming languages like Visual Basic or Java. However, most DI and BI tools have support for SQL override, allowing data transformations to be defined as the full SQL executed as is by the original database. Note that this SQL parsing does not support the syntax of special data stores such as MOLAP Cubes often seen in BI tools such as Oracle Hyperion Essbase.

The section explains the general limitations in expression parsing, in particular with respect to the SQL syntax variations from the SQL 99 standard to the limitations in proprietary variations and database extensions for each popular database such as IBM DB2, Microsoft SQL Server, Oracle, Teradata, etc.

Furthermore, the support for SQL use cases from Data Definition Language (DDL) to Data Manipulation Language (DML) is increasingly limited by the complexity as the SQL features:

1. SQL Views

- SQL Views are defined by a single SQL statement. SQL Views are used to read databases and are therefore mostly based on a complex SELECT statement.
- SQL Views are well supported with only a few limitations based on the some proprietary extensions of some databases.

2. SQL Operations (Stored Procedures, Functions and Triggers)

SQL Operations are defined by multiple SQL statements that are pre-compiled and stored on the server for reuse, and can have parameters. SQL Operations often contain multiple DML statements (SELECT, INSERT, UPDATE, DELETE, etc.), and use control statements (IF, THEN, ELSE, CASE, FOR, etc.). Support for SQL Operations is limited to functions and stored procedures with the following support and limitations:

- The general support for Stored Procedures is defined as follows:
 - Stored procedures reading from and writing to tables (catalog objects): supported
 - Stored procedures lineage support for IN, OUT, INOUT parameters: only IN parameters are supported
 - Stored procedures with cursors: supported
 - Stored procedures with temporary/Local variables: supported
 - Stored procedures using functions: limited support
 - Stored procedures using public/private synonyms: supported
 - Stored procedures using other stored procedures: limited support
 - Stored procedures using temporary tables: supported
 - Stored procedures with error management: not relevant for data flow lineage
- Parsing of independent SQL Stored Procedures and Functions is currently limited to the following database technologies:
(although some statements specific to database proprietary SQL syntax may not be supported)
 - Microsoft SQL Server Database
 - Oracle Database
 - PostgreSQL Database
 - Pivotal GreenPlum Database
 - Teradata Database
- Most dynamic and chaining aspects of the operations are not supported such as:
 - operations calling other operations
 - operations that execute dynamic SQL (e.g. SQL text passed as parameter)
 - external operations (e.g. includes)
- Some usage aspects of such operations by DI and BI tools might be limited:
 - using SQL operations to read data is supported in most DI and BI tool import bridges
 - using SQL operations to write data is supported only in DI tool import bridges such as Informatica PowerCenter, Microsoft SSIS and Talend

3. SQL DML (DI/ETL) Scripts

SQL DML Script files usually consist of multiple SQL statements and sometimes designed to be executed from outside the database, and therefore not necessarily compiled in the database. Such scripts are often handcrafted and sometimes generated or part of the database utilities (e.g. Teradata fastload, bulkload and multiload)

Parsing of independent SQL script files (.sql) is currently limited to the following database technologies:

- Apache Hadoop Hive Database (Cloudera, Hortonworks, MapR) SQL DML (DI/ETL) Script (HiveQL)
- Apache Sqoop
- IBM DB2 Universal Database (UDB) SQL DML (DI/ETL) Script
- Microsoft SQL Server Database SQL DML (DI/ETL) Script (Transact-SQL)
- Oracle Database SQL DML (DI/ETL) Script (PL/SQL)
- PostgreSQL Database SQL DML (DI/ETL) Script
- Teradata Database SQL DML (DI/ETL) Script (BTEQ, Fastload, etc.)

SQL Expression Analyzer Based Import Bridges

MIMB bridges depending on "Transformation Expression Parsing" include the following types of import bridges:

- Database Servers (via JDBC), such as Oracle Database Servers
- Database SQL DDL Scripts, such as Oracle Database SQL DDL Script
- Database SQL DML (DI/ETL) Scripts, such as Oracle Database SQL DML (DI/ETL) Script (PL/SQL)
- Data modeling tools (for SQL View), such as Erwin Data Modeler
- Data Integration (DI) tools (for ETL Transformations), such as Informatica PowerCenter
- Business Intelligence (BI design, OLAP, reporting) tools, such as SAP BusinessObjects Designer
- Metadata Management (MM tools), such as Apache Atlas

SQL Expression Analyzer intrinsic limitations

Unsupported features

- Control lineage is not available for certain expressions inside operations.

Limitations on SQL99 Keywords

The SQL99 syntax includes reserved keywords (e.g. CREATE that cannot be used as a table name), and non reserved keywords. Some of these keywords are used for compound terms like "order by", "group by", "union join". All SQL99 keywords are properly parsed and will not fail the expression analyzer used in bridges. However, some keywords have a direct data flow impact on lineage, and some do not. The following list provide the details:

The SQL99 syntax includes reserved keywords (e.g. CREATE that cannot be used as a table name), and non reserved keywords. Some of these keywords are used for compound terms like "order by", "group by", "union join". All SQL99 keywords are properly parsed and will not fail the expression analyzer used in bridges. However, some keywords have a direct data flow impact on lineage, and some do not. The following list provide the details:

- SQL99 Reserved Key Words

- with direct data flow lineage impact: ADD, ALL, ALTER, AND, ANY, AS, ASC, BEGIN, BOTH, BY, CALL, CASE, CAST, CHAR, CHARACTER, CONSTRAINT, CREATE, CURRENT, CURSOR, DATE, DAY, DEC, DECIMAL, DEFAULT, DELETE, DESC, DISTINCT, DOUBLE, DROP, ELSE, END, EXCEPT, FALSE, FLOAT, FOR, FOREIGN, FROM, FULL, FUNCTION, GRANT, GROUP, HAVING, HOUR, IN, INNER, INSERT, INT, INTEGER, INTERSECT, INTERVAL, INTO, IS, JOIN, KEY, LEADING, LEFT, MINUTE, MONTH, NATURAL, NO, NOT, NULL, NUMERIC, ON, OR, ORDER, OUT, OUTER, PRECISION, PRIMARY, PROCEDURE, RECURSIVE, REFERENCES, RETURN, REVOKE, RIGHT, ROW, ROWS, SECOND, SELECT, SET, SMALLINT, TABLE, THEN, TIME, TIMESTAMP, TO, TRAILING, TRUE, UNION, UNIQUE, UPDATE, USING, VALUES, VARCHAR, VARYING, VIEW, WHEN, WHERE, WITH, YEAR
- with no lineage impact: ABSOLUTE, ACTION, ADMIN, AFTER, AGGREGATE, ALIAS, ALLOCATE, ARE, ARRAY, ASSERTION, AT, AUTHORIZATION, BEFORE, BINARY, BIT, BLOB, BOOLEAN, BREADTH, CASCADE, CASCADED, CATALOG, CHECK, CLASS, CLOB, CLOSE, COLLATE, COLLATION, COLUMN, COMMIT, COMPLETION, CONNECT, CONNECTION, CONSTRAINTS, CONSTRUCTOR, CONTINUE, CORRESPONDING, CROSS, CUBE, CURRENT_DATE, CURRENT_PATH, CURRENT_ROLE, CURRENT_TIME, CURRENT_TIMESTAMP, CURRENT_USER, CYCLE, DATA, DEALLOCATE, DECLARE, DEFERRABLE, DEFERRED, DEPTH, Deref, DESCRIBE, DESCRIPTOR, DESTROY, DESTRUCTOR, DETERMINISTIC, DICTIONARY, DIAGNOSTICS, DISCONNECT, DOMAIN, DYNAMIC, EACH, END-EXEC, EQUALS, ESCAPE, EVERY, EXCEPTION, EXEC, EXECUTE, EXTERNAL, FETCH, FIRST, FOUND, FREE, GENERAL, GET, GLOBAL, GO, GOTO, GROUPING, HOST, IDENTITY, IGNORE, IMMEDIATE, INDICATOR, INITIALIZE, INITIALLY, INOUT, INPUT, ISOLATION, ITERATE, LANGUAGE, LARGE, LAST, LATERAL, LESS, LEVEL, LIKE, LIMIT, LOCAL, LOCALTIME, LOCALTIMESTAMP, LOCATOR, MAP, MATCH, MODIFIES, MODIFY, MODULE, NAMES, NATIONAL, NCHAR, NCLOB, NEW, NEXT, NONE, OBJECT, OF, OFF, OLD, ONLY, OPEN, OPERATION, OPTION, ORDINALITY, OUTPUT, PAD, PARAMETER, PARAMETERS, PARTIAL, PATH, POSTFIX, PREFIX, PREORDER, PREPARE, PRESERVE, PRIOR, PRIVILEGES, PUBLIC, READ, READS, REAL, REF, REFERENCING, RELATIVE, RESTRICT, RESULT, RETURNS, ROLE, ROLLBACK, ROLLUP, ROUTINE, SAVEPOINT, SCHEMA, SCROLL, SCOPE, SEARCH, SECTION, SEQUENCE, SESSION, SESSION_USER, SETS, SIZE, SOME, SPACE, SPECIFIC, SPECIFICTYPE, SQL, SQLEXCEPTION, SQLSTATE, SQLWARNING, START, STATE, STATEMENT, STATIC, STRUCTURE, SYSTEM_USER, TEMPORARY, TERMINATE, THAN, TIMEZONE_HOUR, TIMEZONE_MINUTE, TRANSACTION, TRANSLATION, TREAT, TRIGGER, UNDER, UNKNOWN, UNNEST, USAGE, USER, VALUE, VARIABLE, WHENEVER, WITHOUT, WORK, WRITE, ZONE

- SQL99 Non Reserved Key Words
 - with direct data flow lineage impact: BETWEEN, CONVERT, EXISTS, NUMBER, OVERLAPS
 - with no lineage impact: ABS, ADA, ASENSITIVE, ASSIGNMENT, ASYMMETRIC, ATOMIC, AVG, BIT_LENGTH, BITVAR, C, CALLED, CARDINALITY, CATALOG_NAME, CHAIN, CHAR_LENGTH, CHARACTER_LENGTH, CHARACTER_SET_CATALOG, CHARACTER_SET_NAME, CHARACTER_SET_SCHEMA, CHECKED, CLASS_ORIGIN, COALESCE, COBOL, COLLATION_CATALOG, COLLATION_NAME, COLLATION_SCHEMA, COLUMN_NAME, COMMAND_FUNCTION, COMMAND_FUNCTION_CODE, COMMITTED, CONDITION_NUMBER, CONNECTION_NAME, CONSTRAINT_CATALOG, CONSTRAINT_NAME, CONSTRAINT_SCHEMA, CONTAINS, COUNT, CURSOR_NAME, DATETIME_INTERVAL_CODE, DATETIME_INTERVAL_PRECISION, DEFINED, DEFINER, DISPATCH, DYNAMIC_FUNCTION, DYNAMIC_FUNCTION_CODE, EXISTING, EXTRACT, FINAL, FORTRAN, G, GENERATED, GRANTED, HIERARCHY, HOLD, IMPLEMENTATION, INFIX, INSENSITIVE, INSTANCE, INSTANTIABLE, INVOKER, K, KEY_MEMBER, KEY_TYPE, LENGTH, LOWER, M, MAX, MIN, MESSAGE_LENGTH, MESSAGE_OCTET_LENGTH, MESSAGE_TEXT, METHOD, MOD, MORE, MUMPS, NAME, NULLABLE, NULLIF, OCTET_LENGTH, OPTIONS, OVERLAY, OVERRIDING, PASCAL, PARAMETER_MODE, PARAMETER_NAME, PARAMETER_ORDINAL_POSITION, PARAMETER_SPECIFIC_CATALOG, PARAMETER_SPECIFIC_NAME, PARAMETER_SPECIFIC_SCHEMA, PLI, POSITION, REPEATABLE, RETURNED_LENGTH, RETURNED_OCTET_LENGTH, RETURNED_SQLSTATE, ROUTINE_CATALOG, ROUTINE_NAME, ROUTINE_SCHEMA, ROW_COUNT, SCALE, SCHEMA_NAME, SECURITY, SELF, SENSITIVE, SERIALIZABLE, SERVER_NAME, SIMPLE, SOURCE, SPECIFIC_NAME, SIMILAR, SUBLIST, SUBSTRING, SUM, STYLE, SUBCLASS_ORIGIN, SYMMETRIC, SYSTEM, TABLE_NAME, TRANSACTIONS_COMMITTED, TRANSACTIONS_ROLLED_BACK, TRANSACTION_ACTIVE, TRANSFORM, TRANSFORMS, TRANSLATE, TRIGGER_CATALOG, TRIGGER_SCHEMA, TRIGGER_NAME, TRIM, TYPE, UNCOMMITTED, UNNAMED, UPPER, USER_DEFINED_TYPE_CATALOG, USER_DEFINED_TYPE_NAME, USER_DEFINED_TYPE_SCHEMA

Limitations on Database Extensions

In addition to the list of SQL99 keywords, the expression analyzer has been enhanced to support common database specific constructs. The limitations are in the list below:

- IBM DB2 extension limitations:
 - The CONCAT keyword in DB2 can be used as a function (supported) or as an operator (not supported)

```
SELECT CONCAT(first_name, last_name) FROM customer;
SELECT first_name CONCAT last_name FROM customer;
```

- Oracle extension limitations:
 - Support for Oracle Text Literals is limited to the two single quotation marks mechanism, like:

```
'Jackie''s raincoat'
```

but not the Q/q alternative quoting mechanism, like:

```
Q'Jackie's raincoat>'
```

Limitations on Data Type Support

The expression analyzer doesn't support user defined complex data types.

Conversions of Transformation Expressions in Export Bridges

Some MIMB Export Bridges depend on "Transformation Expression Conversions" to restore the transformation semantic (ClassifierMap and FeatureMap). Such conversion might be limited by the complexity of the transformation expression. If the target tool accepts expressions in textual form, such expressions too complex to restore are converted only in textual form, as defined in the source tool language. If the expression can be successfully restored, it is transformed into the target tool language.

MIMB bridges depending on "Transformation Expression Conversions" include the following export bridges:

- Data Integration (DI) tools (for ETL Transformations), such as:
 - Informatica PowerCenter
 - IBM DataStage
 - Microsoft SQL Server Integration Services (SSIS)
 - Oracle Data Integrator (ODI)
 - Oracle Warehouse Builder (OWB)
 - Talend DI
- Business Intelligence (BI) design tools, such as:
 - IBM Cognos Framework Manager (FM)
 - Microsoft SQL Server Analysis Services (SSAS)
 - SAP BusinessObjects Designer
 - SAP BusinessObjects Information Design Tool

Internationalization Aspects

The internationalization of the MIMB bridge names, parameters, tooltips and messages is managed by XML files within `[MIMB-INSTALL-DIR]\conf\MIRI18n\`

The support for transferring internationalized (e.g. multi-byte character sets) through MIMB is achieved via the UTF8 encoding in the MIR neutral metamodel. The MIMB import and export bridges which cannot automatically detect the encodings of the tools, offer a parameter called "Encoding" allowing the user to specify one. There are however a few minor limitations:

- API based bridges running on Windows, if the tool only works for the LOCALE encoding of the machine. In such case we only support the "English" API while the vendor may offer a different tool and therefore API for other languages. Such API based tools are very limited and include:
 - CA Gen (previously COOL:gen)
- Java based bridges running on Windows, if the JDK installation path is not expressed in the LOCALE encoding of the machine (due to an outstanding long time bug in most popular JVM implementations). Such Java based bridges include:
 - IBM Cognos ReportNet Repository
 - IBM InfoSphere Data Architect (previously Rational Data Architect)
 - IBM Rational Software Architect
 - IBM WebSphere Metadata Server
 - Oracle Business Intelligence (previously Siebel Analytics) Enterprise Edition
 - Oracle Data Integrator
 - Oracle Warehouse Builder
 - SAP BusinessObjects Crystal Reports
 - SAP BusinessObjects Information Design Tool
 - SAP BusinessObjects Web Intelligence
 - SAP BusinessObjects Repository
 - SAP Business Warehouse
 - SAS Business Intelligence
 - Tableau
- File based bridges reading or writing Unicode files in 16 bits fixed size encoding which is extremely rare.

Multi-Model Import Bridges for Incremental Metadata Harvesting

Import bridges can be used for metadata harvesting into the repository of a Metadata Management (MM) solution. The metadata to be harvested can be very large, especially from Business Intelligence (BI) environments which may contain thousands of reports. Consequently some import bridges are implemented as a "Multi Model" bridge managing the large amount of metadata in multiple Model files. This approach also allows for incremental harvesting where the import bridge can check what has changed since the last import date, and return an updated "table of contents" indicating which models have been added, modified or deleted and import only the new or updated models. Such feature is supported only by a limited amount of import bridges listed below:

- erwin Data Modeler's Mart
- IDERA ER/Studio Repository
- IBM Cognos Content Manager
- Microsoft SQL Server Analysis and Reporting Services (SSAS, SSRS)
- MicroStrategy Intelligence Server
- Oracle Business Intelligence(Siebel Analytics) Enterprise Edition
- QlikTech QlikView
- SAP BusinessObjects Repository
- SAS Business Intelligence
- Tableau

Linux vs. Windows Only Availability

Most bridges are written Java and can therefore run on both Windows and Linux platforms where a Java RunTime Environment (JRE) is available. However, a significant amount bridges are written in C++ compiled only for Windows, especially when they depend on Windows only COM API, such as:

- erwin Data Modeler binary native file formats (Windows only COM API).
- SAP Business Objects Universe Designer (Windows only COM API).
- Microstrategy Intelligence Server (Windows only COM API).

The bridges running only on Windows are annotated with a Windows icon on the published supported tools at: <http://www.metaintegration.net/Products/MIMB/SupportedTools.html>.

Third-party Software (drivers) Internet Download

Some bridges requiring internet access to <https://repo.maven.apache.org/maven2/> (and exceptionally a few other tool sites) in order to download the necessary third party software drivers into \$HOME/data/download/MIMB/ (such directory can be copied from another MIMB server with internet access).

The bridges requiring internet to download third party software drivers are annotated with a Download icon on the published supported tools at: <http://www.metaintegration.net/Products/MIMB/SupportedTools.html>.

Within that context, here are the known limitations.

On Windows server 2016 (or newer), a message "Cannot execute hadoop.dll" can appear in the following bridges:

- Amazon Web Services (AWS) S3 Storage (Import)
- Apache Hadoop Distributed File System (HDFS Java API) (Import)
- Apache Kafka (Import)
- Excel File (XLSX) (Import)
- File System (CSV, Excel, XML, JSON, Avro, Parquet, ORC, COBOL Copybook) (Import)
- Flat File (CSV or Excel) - Beta Bridge (Import)
- Metadata Excel Format (Export)
- Metadata Excel Format (Import)
- Microsoft Azure Blob Storage (Import)
- Microsoft Azure Data Lake Storage (Import)
- OpenStack Swift Object Store - New Beta Bridge (Import)
- Oracle Cloud Object Storage Classic - New Beta Bridge (Import)

That is because the official hadoop.dll downloaded from the Maven repository has a dependency to the old Windows onmsvcr100.dll no longer delivered with the new Windows 2016 Server. The solution is add the missing C runtime msvcr100.dll.of by adding the [Microsoft Visual C++ 2010 Service Pack 1 Redistributable Package MFC Security Update](#).