Caché SQL — Version F.14
Release Information

Version: Caché SQL F.14

Date: June 25, 1998

Part Number
IS-SQL-0-F.14A-CP-R

Caché SQL — F.14 Release Information

For support questions about any InterSystems products, contact the InterSystems Worldwide Response Center:

Phone: US: +1 617 621-0700 Europe: +44 (0) 1753 830-077
Fax: US: +1 617 374-9391 Europe: +44 (0) 1753 861-311
Internet — support@intersys.com **FTP Site** -— ftp.intersys.com
World Wide Web — www.intersys.com

BBS: General Use: +1 (617) 225-0475 Europe: +44 (0) 1753-853-534

Table of Contents

APPENDIX

A

Caché SQL Version F.14 Release Information

New Features

Year 2000 Support

- Caché SQL version F.14 includes year 2000 support via four digit year displays on all form windows. Four digit dates are displayed depending on user settings and on the version and platform of Caché being run. General rules for date display are as follows:
  - If Caché is started, and the sliding window is not set, years in the 20th century display as two digits. Years not in the 20th century display as four digits. (Note: on the ISM VAX product, if the default window is not set, it defaults to the current century, as determined from $H.)
  - If the sliding window is set, then years in the sliding window display as two digit, and years outside the window display as four digits.
  - If the sliding window is cleared by a commandsuch as: s x=$$EmptyProcWindow^%DATE then all dates display as four digits. Attempts to enter a two digit date result in an error.

Note: For more information on setting the sliding window, see your Caché version 6.4 release information.

Installation

- F.14 has 2 versions — One for Caché and one for non-Caché platforms (ISM / DSM / DTM). Caché has an installation option to select the dbms version you want to install; G or F.14. The “G” version contains object SQL capabilities. The F.14 version has a server “switch” to turn on the ‘old’ server.

Macro Compiler

- Caché supports language modes for MACRO routines. This allows you to:
  - Switch language modes when using the full screen editor.
  - Compile and copy routines in a specified language mode.
  - Output and input routines in specified language mode. Routine input preserves the language mode specified in the routine output. If no language mode is specified, you are for the language mode.
  - Display language mode in detailed Routine Display. For Caché language mode, the field is left blank.

Client/Server

- There are two new functions to convert FileMan Date/Time Stamps between external and internal values, FMdtint and FMdext. These functions ensure that the appropriate ODBC format is sent to the client for FileMan Date/Time [stamp] fields.

  Sample Field Validation Code using the new Fileman date/time conversion functions:

```
Internal to External M Code (1/2)
 s %val=$$FMdtext^%qarzdt(%val,"S",0
 s:%val=-1 %val="",%ok=0
 External to Internal M Code (0/2)
 s %val=$$FMdtint^%qarzdt(%val,"S",0)
 s:%val=-1 %val="",%ok=0
```

  Note: You must delete all stored procedures on the server after installing F.14.

- In an effort to make our DDL create fields more compatible with ODBC, the following changes have been made to DDL

  - If a field is created with an SQL SMALLINT datatype, the field has the following characteristics:

    Maximum Length = 6
    Maximum Value = 32768
    Minimum Value = -32768

  - If a field is created with the SQL INTEGER (or INT) datatype, the field has the following characteristics:

    Maximum Length = 11
    Maximum Value = 2147483648
    Minimum Value = -2147483648

- The TCOMMIT function now checks to make sure that a process is in a transaction before a TCOMMIT is issued.

  Sample TCOMMIT statement:

```
i $$intp^%ratp TCOMMIT s SQLCODE=$zu(34)
```

- Caché SQL ODBC Driver (version 2.20.0009 and greater) has username/password encryption capability. If you want to use encryption, define the ENCRYPT flag in the registry with a non-zero value. If ENCRYPT is defined, username/password encryption is used during the initial client/server handshake.

Utilities

- Command level utilities display all % routines in implied namespaces when you specify * at the Routine(s): prompt. You still must specify %* to see % routines in mapped namespaces.

Import

- InterSystems has added a new API for “intelligent” export/import. This feature allows you to import data into your database without overwriting existing data. Intelligent import is called via the following extrinsic function:

```
s result=$$import^%msqlapi(filename,validate,killglo,abnormal,compile,local
```

Table A-1: $$import^%msqlapi parameters

Field Name	Description
filename	The input device (typically a flat file) from which you want to import the ^mx globals. A null value assumes import from ^mx... global in your current directory.
validate	Specify the method used to handle objects already in the target directory that are dependent on imported objects but are not included in the import list themselves. For example, if the target directory contains base table “ABC” and dependent form “qed”, while the import list includes base table “ABC” but not form “qed”, form “qed” constitutes a dependent object that is handled subject to the option specified in this field. Options: - **1** — **Revalidate** (default). This option scans the current directory for all objects that are dependent on imported objects but not included in the import list themselves, tracing dependency back as many generations as it goes. It then adds these dependent objects to the import list by externalizing them into the ^mx globals. When it outputs the ^mx globals, it overwrites the old objects with the imported objects, thus revalidating them. This is the most reliable way to ensure the integrity of the import directory. - **2** — **Prompt.** This option lets you choose which objects to revalidate and which to delete. It builds a master list of all dependent objects not in the import group and displays the list in an interactive window. You can navigate through the list and selectively revalidate or delete these objects. - **3** — **Delete.** This option automatically deletes all objects that are dependent on imported objects but not included in the import list themselves, tracing dependency back as many generations as it goes. When you use the *Delete* option, the Import Results report shows a list of all dependent objects that are deleted from the target directory.
compile	- **Y — Yes**. Automatically compile all imported base tables, forms, reports, queries, and menu objects after they are imported into the target directory. *Yes* is the default response. - **N — No**. Leave objects uncompiled. **Note:** The import run takes considerably longer when you recompile all imported objects.

Table A-1: $$import^%msqlapi parameters (Continued)

Field Name	Description
killglo	Specify how you want to handle the ^mx globals after successful completion of the import (i.e., without abnormal termination). When the import procedure successfully scans through all the objects in the export list and completes all the necessary steps, this is considered to be a "normal" end, even if some objects incurred errors during the load or compile phase. Options: ▫ **P** — **Preserve all.** This option preserves all information in the ^mx globals. ▫ **K** — **Kill all.** This option purges all information from the ^mx globals. ▫ **S** — **Kill on success.** This option purges the ^mx globals of all objects that successfully loaded and compiled (if the compile option is enabled) and preserves any objects that did not load and compile correctly. If you import different groups of objects from the same device, you must kill the ^mx globals between each import set. In general, it is a good idea to do this all the time, unless you have some explicit reason for saving them. If you are importing globals from a flat file, the default response for this field is *Kill all*. If you are not importing globals from a flat file, the default response for this field is *Preserve all*.
abnormal	▫ **Y — Yes**. purge the ^mx globals after abnormal termination of the import run. ▫ **N — No (default)**. preserve the ^mx globals. An "abnormal" end means that the import procedure was interrupted by an event that prohibited it from scanning through all the objects in the export list (e.g., <CTRL-C>)
local	▫ **N** — **No (default)**. Caché stores the temporary import tables in globals. This protects against <STORE> errors but causes the import to run more slowly. ▫ **Y** — **Yes**. Caché stores the temporary import tables in local arrays. This makes the import run faster. If you have a substantial amount of local memory and the target directory does not contain a large number of objects, you can safely use local memory. If, however, you encounter a <STORE> error while importing in local memory, you should change your response to *No*. **Caution:** If you have less than 256K of local memory, you should always answer *No*.

The import set name assigned when an import is invoked using the api consists of the file name, if from a flat file, or mxdd, if from an existing mxdd; date; time the api was invoked; and the word Import.

Both Report on Import and Detailed Report on Import may also be run for api generated import sets.

For example, an import done from a flat file called isc.exp would have a import set name of "isc.exp 09/24/96 3:10PM import". The import results can be examined by going into the import run option selecting the import set name automatically created by the api and pressing return on the **See Import Results?** branching button.

- Caché SQL allows General Imports from an ^mxdd file created using previous versions of M/SQL. Formerly, you received a verison mismatch error and the import aborted. You now receive a warning message about the version mismatch and the import continues. The success of this operation depends on the version you are importing from.

DDL

- Caché SQL now supports the DDL syntax described below:

LONGVARCHAR ['(' <length> ')'] as a <character string datatype>.

ALTER TABLE <table name> ADD FOREIGN KEY ['('] <referencing columns> [')'] <referencing specification>

ALTER TABLE MODIFY [COLUMN] ...
As an alias for ALTER TABLE ALTER [COLUMN] ...

ALTER TABLE ALTER [COLUMN] <column name> <alter column required constraint>
Where: <alter column required constraint> ::= NULL | NOT NULL
This can be used to make a field required or not required.

ALTER TABLE ALTER [COLUMN] <column name> <datatype>
As a way to change a column's datatype

CREATE UNIQUE INDEX ...
As an alias for CREATE INDEX ...

Note: In the 'ON TABLE' clause in both the CREATE INDEX and DROP INDEX statements, the word TABLE is now optional.

CREATE/DROP VIEW <schema>.<viewname>

CREATE/DROP INDEX <schema>.<indexname>

ALTER TABLE <tablename> ADD FOREIGN KEY <constraint name> ['('] <referencing columns> [')'] <referencing specification>

Queries

- Caché SQL now accepts multi-field outer-join statements. You can only join more than two tables if you are joining on the row id field.

```
SELECT field,....
FROM (Table1 LEFT JOIN Table2 on Table1.rowID=Table2.rowID) and
        (Table2 LEFT JOIN Table3 on Table2.rowID=Table3.rowID)
```

The statement above is equivalent to:

```
SELECT field,...
FROM Table1, Table2, Table3
WHERE (Table1.rowID=Table2.rowID) and (Table2.rowID=Table3.rowID)
```

- Caché SQL provides several performance improvements via optimization of SELECT queries using the DISTINCT keyword with the GROUP BY clause.
- Caché SQL allows you to exclude certain fields from a SELECT * query. If you answer ***Yes*** to "Exclude Field From Select * ?" in the Additional Options menu bar on the Field Definition form, the current field is eliminated from all SELECT * queries.

Unicode

- Wide characters (i.e. Kanji) specified in the NLS "IDENT" tables, can now appear within identifiers (such as table and field names) in SQL queries.
- Caché Developer forms now accept wide characters (i.e. Kanji characters) for platforms that support NLS and wide characters.

F.14 Corrections

SQL

- InterSystems has corrected an <UNDEFINED> error with a query based on a view where the field used in the condition had an index option.
- InterSystems has corrected a syntax error in LEFT OUTER join statements when the WHERE condition was fully contained within parenthesis.
- InterSystems has corrected a <SUBSCRIPT> error in queries of the following form:

```
DELETE FROM My_Table
WHERE My_Field = 0 OR My_Field is NULL
```

  where My_Table has 2 maps:

```
^glo("master",{primary_key})={My_Field}    <-- Master Map
^glo("index",{My_Field},{primary_key})=""  <-- Index Map
```

- InterSystems has corrected an <UNDEFINED> error processing queries with an IN comparison against a field that was part of an index map.
- InterSystems has corrected a problem that caused long queries to be missing characters. This problem caused “Field not found”, or “Table not found” errors trying to run queries over 480 characters in length.
- InterSystems has corrected a problem that caused SELECT queries with invalid field names to report back an error without the name of the invalid field.
- InterSystems has corrected an <UNDEFINED> error compiling a query joining two views that each have calculated fields with the same alias (e.g. select field1+field2 as total) on both views). Example:

```
View1: SELECT Field_1 + Field_2 AS Total
        FROM My_Table
View2: SELECT Field_1 + Field_2 AS Total
        FROM My_Table
Query: SELECT View1.Total, View2.Total
        FROM View1, View2
```

- InterSystems has corrected a problem where aliasing general views caused incorrect query parsing.

Unicode

- InterSystems has corrected a problem where wide characters were not displayed properly in lookup boxes on forms.
- InterSystems has corrected a problem in the Data Import Utility which caused imports with Field Order set to 'User Defined' not to work.

DSM

- InterSystems has corrected an <UNIMPLEMENTED> error compiling routines with embedded SQL on DSL platforms. The directory reference now accepts two comma pieces to support DSM's UCI,VOL directory specification.

Client/Server

- Rowid fields for tables created with default structure (including those created via DDL) are automatically hidden from ODBC clients.

  Any fields designated hidden are not displayed on a query of the form SELECT * FROM TABLE, however, you can explicitly select hidden columns by name.

  Also, any fields designated hidden cannot have an INSERT or UPDATE operation performed on them unless they are explicitly specified in the column list. If no column list is specified, you may not provide values in the VALUES clause for hidden fields. Any mismatch between the number of values in the VALUES clause and the number of non-hidden columns generates an SQL error.

- A problem has been corrected where corrupt data on the server could cause date fields on the client to have 1/1/1841 values. They are now sent to the client as null values.

- The ODBC Prepare function now correctly describes the ODBC datatype of numeric fields, based on the field's defined properties - length, #decimals, minval, and maxval. The possible datatypes are:

datatype	return value	description
Small Integer	5	Integers between -2^{15} and $2^{15}-1$ inclusive i.e. between -32768 and 32767
Integer	4	Integers between -2^{31} and $2^{31}-1$ inclusive i.e. between -2147483648 and 2147483647
Double	8	All other numeric fields

- InterSystems has corrected a problem with Fileman date and time values sent to client applications.

- InterSystems has corrected a problem inserting new records through Microsoft Access. Subsequent INSERT calls had improper or non-existent insert values in the %oms(%omsacn) array.

- InterSystems has corrected an <UNDEFINED> error running the server in DEBUG mode.

Import/Export

- You can now have field names the same as the base table name when creating tables through DDL.
- InterSystems has corrected a problem in the Data Import Utility which caused imports with the field order set to 'User Defined' not to work.

Queries

- InterSystems has corrected an <UNDEFINED> error running a query with a constraint against an indexed ALPHAUP name datatype field.

Available Documentation

Caché SQL

The documentation set for InterSystems Caché SQL relational database product includes the following manuals:

- This *Caché SQL — Version F.14 Release Information*; Revision Date: May 1998
- *Open M with SQL — Version F.12 Release Information*; Revision Date: September, 1997.
- *Open M/SQL Developer Guide* — Version F.6 & F.7; Revision Date: September 11, 1995.
- *Open M with SQL Database Administrator's Guide — Version F.9, F.10;* Revision Date: December 9, 1996.
- *User Interface Programming Guide — Version F.4;* Revision Date: October 6, 1994.
- *Open M with SQL Data Dictionary Guide* — Version F.10; Revision Date: April 2, 1997.
- *Open M/SQL M/PACT (*includes *M/PACT Addendum)* — Version B; Revision Date: July 2, 1990.

Caché SQL Server

If you plan to use Caché SQL Server, you also need the following documentation:

- *Caché SQL Driver User's Guide* —Version 2.2; Revision Date: September, 1997.
- *Open M Relational Server Manager's Guide* — Version F.10; Revision Date: December 9, 1996.

For Caché, the two books above have been combined into the following text:

- *Caché SQL Connectivity Guide* — Version 3.1; Creation Date: September 1998