Open M with SQL — Version F.15
Release Information

Version: Open M with SQL F.15

Date: March 9, 2000

Part Number
IS-SQL-0-F.15A-CP-R

Open M with SQL Version F.15 Release Information

For Support questions about any InterSystems products, contact the InterSystems Worldwide Support 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
E-mail:	support@intersys.com	**FTP site:** ftp.intersys.com
World Wide Web:	www.intersys.com	
BBS:	Europe: +44 (0) 1753-853-534	

Table of Contents

Open M with SQL Version F.15 Release Information

A Converting Your Applications to the New Global Structure for Routines

Open M with SQL Version F.15 Release Information

Open M with SQL Version F.15 runs on Caché 3.x systems. It can be used instead of Caché SQL for compatibility with legacy systems. These release notes detail the differences between F-15 and previous releases of the F DBMS.

Open M with SQL Version F.15 uses the same ODBC client / server configuration as Caché SQL. For information about ODBC client / server configuration and connectivity, see the *Caché ODBC Driver Guide*.

Converting Your Applications to Open M with SQL Version F.15

When upgrading to Open M with SQL version F.15, perform the following steps to convert your existing Open M with SQL applications:

1. Back up your system. Once you have converted to the new routine storage global structure, you cannot downgrade to earlier versions of Open M with SQL.

2. Open M with SQL provides the Conversion Manager utility to assist you in running the appropriate conversion program(s).

   Run the Open M with SQL Conversion Manager utility in each namespace using the FDBMS database as the Caché library (except the %FDBMS namespaces). The Conversion Manager consists of a series of routines that update certain Open M with SQL internal structures to make them compatible with the new version.

3. After running the conversion routines, you should recompile all Open M with SQL objects in each namespace using the FDBMS as the Caché library (except the %FDBMS namespace).

   Open M with SQL provides the %mcompil utility to assist you in recompiling Open M with SQL objects.

Note: All Stored Procedures must be deleted from the Caché Server prior to connecting from the Caché SQL Driver.

Routine Global Conversion

If you are upgrading from version F.10 or earlier, you must run the ^ROUTINE global conversion utility. SeeAddendum A, *Converting Your Applications to the New Global Structure for Routines* for information on converting your applications to the new global structure for routines.

Warning: You must convert applications to the new global structure for routines prior to running Open M with SQL version F.15 or greater. Serious database problems may occur if this conversion is not run.

Conversion Manager Utility

The Conversion Manager utility runs conversion routines beneath a window-based interface. It automates the task of converting to a new Open M with SQL version on a per namespace basis and allows you to selectively exclude undesired features of the target version. The Conversion Manager also allows you to run conversion routines multiple times and print reports associated with a particular conversion task.

You should normally run the Conversion Manager in each namespace that you are upgrading.

For complete information on how to convert Open M with SQL applications using the Conversion Manager utility and recompile Open M with SQL objects using the %mcompil utility, see the *Open M with SQL Database Administrator's Guide*.

F.10 to F.11 Conversion Tasks

Once the routine conversion is complete, run the F.10 to F.11 conversion tasks from the Conversion Manager Menu:

```
                          Conversion Manager
NOTE:  Some conversions may not be necessary if they do not apply to a
       DBMS software upgrade.

From Version           To Version
B                      C                       < Conversion Requirements >
C                      D                       < Conversion Requirements >
D                      E                       < Conversion Requirements >
E                      F                       < Conversion Requirements >
F.*                    F.10                    < Conversion Requirements >
F.10                   F.11                    < Conversion Requirements >
F.12/F.13/F.14         F.15                    < Conversion Requirements >
                                               < Conversion Requirements >

Conversion Manager                                    Press <Help> For Help
```

Table 1: F.10 to F.11 Conversion Task

Conversion Task	Description	Recommendation
Validate & Repopulate General View Fields	This conversion populates new information for existing views which use expressions, aggregates, functions, etc. as column definitions. This provides more accurate information to queries using the Caché SQL ODBC Driver. Additionally, this conversion program checks existing query-based views and returns a report of all views that are incorrectly defined.	Required. You must run this conversion if you are upgrading from an Open M with SQL version prior to F.11

F.12/F.13/F.14 to F.15 Conversion Tasks

InterSystems has added an option for F.12 / F.13 / F.14 to F.15 conversion tasks to the Conversion Manager Menu. This option includes the following conversion task:

Table 2: F.12/F.13/F.14 to F.15 Conversion Task

Conversion Task	Description	Recommendation
Rename System Globals/ Routines	Performs routine/global name changes in Open M with SQL Objects (tables, forms, etc) for routines and/or globals that were referenced from default conversion code.	Required. You must run this conversion for all upgrades.

Installation

F.15 is an optional component of Caché 3.1. By default, Caché 3.1 includes Caché SQL. See your installation guide for information on installing Open M with SQL version F.15 with Caché 3.1.

Routine/Global Name Changes

Many globals and routines have been edited and / or combined to decrease space requirements and improve compilation time in Open M with SQL. Obsolete globals and routines have been removed from the product.

Warning: If you have any direct references to InterSystems system routines and/or globals, they may cause serious errors in your application. This means any globals or routines in the %a*-%y* range. Any user application globals and routines in the %SYS namespace must begin with %z. If you specific questions regarding InterSystems global/routine name changes, please contact InterSystems Worldwide Response Center.

ODBC Support

F.15 product uses the ODBC 3 driver included in your Caché distribution. The F.15 version does not support the following features which are supported by Caché SQL.

- Stored Procedures
- Blobs

In order to initialize both catalog query information in ^%SYS and any existing user defined tables / views for %qserver you must convert your data to ODBC client 3.0.

Procedure **To convert catalog query information nd user defined tables to ODBC 3.x format:**

1. Run the ^%msql routine.
2. Select System Management menu | Conversion Manager menu | F.11 / F.12 / F.13 / F.14 to F.15 Conversion Requirements | Execute Conversion Initialize metadata for ODBC Client 3.0.

New Features

Year 2000 Support

Open M with SQL version F.15 includes year 2000 support. All date displays in the Open M with SQL Developer and the %ur* utility routines have been changed to display the year as 4 digits.

Routine/Global Name Changes

Open M with SQL provides considerable performance improvements. Many globals and routines have been edited and / or combined to decrease space requirements and improve compilation time. Obsolete globals and routines have been removed from the product.

Warning: If you have any direct references to InterSystems system routines and/or globals, they may cause serious errors in your application. This means any globals or routines in the %a*-%y* range. Any user application globals and routines in the %SYS namespace must begin with %z. If you specific questions regarding InterSystems global/routine name changes, please contact InterSystems Worldwide Response Center.

Enhanced SQL System Privileges

Open M with SQL version F.15 provides a mechanism for manipulating user definitions and security settings through ODBC connections and form driven means other then Open M with SQL. The system privileges are only accessible and enforced when using SQL, whether on the server or through ODBC. These new system privileges are accomplished by the product enhancements described below:

Note: The Open M with SQL developer user privileges are not affected by this change and continue to work in the Open M with SQL developer environment.

- Add / Edit / Delete User Definitions

Support has been added for the following DDL Statements:

DDL Syntax	Description
CREATE USER <user> IDENTIFY BY <password>	Creates a new user definition named <user> with a password of <password>
ALTER USER <user> IDENTIFY BY <password>	Changes <user>'s password to <password>
DROP USER <user>	Deletes the user definition named <user>

- System Privilege Definitions

  The following system privilege definitions can now be granted to SQL users:

Privilege	Description
ALTER USER	Allows grantee to alter any user's password
CREATE ROLE	Allows grantee to create roles
CREATE USER	Allows grantee to create users
GRANT ANY PRIVILEGE	Allows grantee to grant any system privilege.
GRANT ANY ROLE	Allows grantee to grant any role
DROP ANY ROLE	Allows grantee to drop roles
DROP USER	Allows grantee to drop users

  These privileges are accessible through extensions to our support for grant and revoke roles.

  System privilege definitions are defined by using the following syntax:

```
<grant role/system privilege statement> ::= GRANT <role/system privilege granted>
                        [ { <comma> <role/system privilege granted> }... ]
                        TO <grantee> [ { <comma> <grantee> }... ]
                            [ WITH ADMIN OPTION ]


<role/system privilege granted> ::=<role name>
                                  | <system privilege>


<system privilege> ::=%ALTER_USER
                      | %CREATE_ROLE
```

```
                    | %CREATE_USER
                    | %GRANT_ANY_PRIVILEGE
                    | %GRANT_ANY_ROLE
                    | %DROP_ANY_ROLE
                    | %DROP_USER


<revoke role/system privilege statement> ::= REVOKE <role/system privilege
revoked>
                    [ { <comma> <role/system privilege revoked> }... ]
                    FROM <grantee> [ { <comma> <grantee> }... ]


<role/system privilege revoked> ::=<role name>
                    | <system privilege>
```

Table 3: Usage rules for system privileges

A system privilege is effectively granted to a user or role only once. One revoke statement removes the privilege from the user..
Unlike revoking privileges on objects definitions, no cascading revokes are performed. For example: User A grants %ALTER_USER to user B WITH ADMIN OPTION and user B grants %ALTER_USER to user C. If user A then revokes %ALTER_USER from user B, user C still has %ALTER_USER privilege.
If you grant a privilege to a user, the user can immediately exercise the privilege.
If you revoke a privilege from a user, the user immediately loses the privilege.
If you grant a privilege to a role, users who have been granted the role can immediately exercise the privilege.
If you revoke a privilege from a role, users who have been granted the role immediately lose the privilege.
If you grant a privilege to PUBLIC, all users can immediately exercise the privilege.
If you revoke a privilege from PUBLIC, all users who were granted the privilege through PUBLIC immediately lose the privilege. If a privilege was granted directly to the user or through a role, the user retains the privilege.
To grant a system privilege, you must either have been granted the system privilege with the ADMIN OPTION or have been granted %GRANT_ANY_PRIVILEGE system privilege.
To revoke any system privilege, you must have been granted the system privilege or role with the ADMIN OPTION.
The REVOKE command can only revoke privileges that have been granted directly with a GRANT statement. The REVOKE command cannot revoke privileges not granted to the revokee or revoke privileges granted to the revokee through roles.

Table 3: Usage rules for system privileges

A grant with the ADMIN OPTION supersedes a previous identical grant without the ADMIN OPTION. If you grant a system privilege to a user without the ADMIN OPTION, and then subsequently grant the privilege to the user with the ADMIN OPTION, the user has the ADMIN OPTION on the privilege or role.
A grant without the ADMIN OPTION does not supersede a previous grant with the ADMIN OPTION. To revoke the ADMIN OPTION on a system privilege from a user, you must revoke the privilege from the user altogether and then grant the privilege to the user without the ADMIN OPTION.

- %CHECKPRIV Enhancements for System Privileges

  %CHECKPRIV enhancements provide a programmatic way to determine if the current user has a system privilege or not.

  The syntax for the new statement is:

```
<checkpriv statement> ::= %CHECKPRIV [ <grant/admin clause> ]
                          <action or system privilege>
                          [ ON [ <object type> ] <object list> ]

<grant/admin clause> ::= GRANT OPTION FOR
                       | ADMIN OPTION FOR

<action or system privilege> ::= <action type> { [ , <action type> ] ... }

<action type> ::= %ALTER
                | SELECT
                | INSERT
                | UPDATE
                | DELETE
                | REFERENCES
                | %ALTER_USER
                | %CREATE_ROLE
                | %CREATE_USER
                | %GRANT_ANY_PRIVILEGE
                | %GRANT_ANY_ROLE
                | %DROP_ANY_ROLE
                | %DROP_USER
<object type> ::= %FORM
                | %MENU
                | %QUERY
                | %REPORT
                | %MENUOBJ
<object list> ::= <object name> [ { , <objects name> ] ... }
<object name> ::= <identifier>
```

  The WITH ADMIN OPTION phrase is optional. If you specify this phrase, the query checks whether or not the user holds the grant privilege on the specified system privilege; not whether or not the user holds the specified system privilege itself.

  If the user is privileged, SQLCODE = 0.
  If the user is not privileged, SQLCODE = 100.

  For Example:

  %CHECKPRIV %CREATE_USER

Returns SQLCODE=0 if the user has privileges to create users. Otherwise, it returns 100.

Installation

- F.15 is an optional component of Caché 3.1. By default, Caché 3.1 includes Caché SQL. See your installation guide for information on installing Open M with SQL version F.15 with Caché 3.1.

Client/Server

- If you install Open M with SQL version F.15 with Caché, the Relational Server is no longer an option in the Open M with SQL developer menu.
- F-15 allows you to execute M code from within the server during the logon process for a server request that been added to the server. You activate on a namespace basis in the namespace using the command

  ```
  s ^mcq("init code")=executable code.
  ```

- Queries via ODBC that use ODBC escape syntax for dates that are against a table with conditional indexes on the data field uses that index.
- Improved handling for OBCS escape sequences.
- The F.15 Relational Server handles encrypted username and password optionally passed by the client. Set the ODBC client ENCRYPT flag in the registry to a non-zero value. If ENCRYPT is defined, username/password encryption is used during the initial client/server handshake.

Procedure **Enabling Username/Password Encryption for the Caché SQL Driver:**

1. Run the Microsoft Registry Editor:

   C:\WINDOWS\REGEDIT.EXE

2. Select HKEY_CURRENT_USER
   Software
   ODBC
   ODBC.INI
   InterSystems

3. Create a new string value, ENCRYPT, and set the value to 1.

SQL

- The default number of days before Cached Queries are purged has been changed from 40 to 7.
- Open M with SQL now supports the following functions with our SQL parser and code generator:

Function	Description
getdate()	Returns the current date and time
datepart(*datepart, date*)	Returns an integer value for the specified part of a date/time value.
datename(*datepart, date*)	Returns the name of the specified part of a date/time value as a character string.
datediff(*datepart, date1, date2)*	Returns date2 - date1, measured in the specified date part.
LTRIM(string-expr)	Returns string-expr with leading blanks removed.
RTRIM(string-expr)	Returns string-expr with trailing blanks removed.

For example:

```
&sql(DECLARE dcur CURSOR FOR
          SELECT Lookup, getdate(*),
                 datepart(month,getdate()),
                 datename(month,getdate()),
               datediff(day,'15 april 1998','17 apr 98'),
                     'z'||LTRIM('  M  ')||'z',
                     'z'||RTRIM('  M  ')||'z'
          INTO :lookup, :a, :b, :c, :d, :e, :f
           FROM Employee)
```

Note: For more information about these functions, see Sybase SQL Anywhere documentation at http://teamserver.icat.com/Sybase.

- Open M with SQL now supports the SQL standard syntax:

  CREATE VIEW <view-name> (<column-list>) AS ...

  The initial <column-list> is a list of AS qualifications for all the SELECT-List Items in the query.

For example:

```
&sql(CREATE VIEW Employee_view_NEW (Emp, Exp1) AS
            SELECT Employee, sal1 + sal2
            FROM Employee)
```

- Open M with SQL now supports the SQL-92 Standard COALESCE Function:

  <coalesce expression> ::=

  COALESCE (<value expression> [,<value expression>]...)

For example:

```
COALESCE (value1, value2, ... valuen)
```

If (value1) is not null, then the value of the COALESCE is (value1).
If (value1) is null, then (value2) is checked.
Once a non-null value (valuex) is found, the value of the COALESCE is set to that value (valuex). If every value, including (valuen) is found to be null, the value of the COALESCE is NULL. The data types of all the values must be comparable.

Note: You may not pass more than 20 values into the COALESCE function.

- InterSystems changed the SQLCODE Value -1017 Invalid Username / Password to the SQLCODE Value -402 Invalid Username / Password.
- This chart shows a listing of SQLCODE values for F-15.

Table 4: The SQLCODE Values for F-15

SQLCODE	Definition
-11002	WinSock: Nonauthoritative host not found
-11001	WinSock: Host not found
-10093	WinSock: Successful WSASTARTUP not yet performed
-10092	WinSock: WINSOCK DLL version out of range
-10091	WinSock: Network subsystem is unavailable
-10070	WinSock: Stale NFS file handle
-10065	WinSock: No route to host
-10064	WinSock: Host is down
-10061	WinSock: Connection refused
-10060	WinSock: Connection timed out

Table 4: The SQLCODE Values for F-15

SQLCODE	Definition
-10058	WinSock: Cannot send after socket shutdown
-10057	WinSock: Socket is not connected
-10056	WinSock: Socket is already connected
-10055	WinSock: No buffer space available
-10054	WinSock: Connection reset by peer (due to timeout or reboot)
-10052	WinSock: Net dropped connection or reset
-10051	WinSock: Network is unreachable
-10050	WinSock: Network is down
-1017	Invalid Username/Password
-500	Fetch row count limit reached
-471	Duplicate cursor name
-470	Option value changed
-469	Driver not capable
-468	Fetch type out of range
-467	Column type out of range
-466	Invalid parameter number
-465	Invalid string or buffer length
-464	Function sequence error
-463	Invalid column number
-462	Memory allocation failure
-461	Communication link failure
-460	General error
-453	Error in User Initialization Code
-452	Message sequencing error
-451	Unable to receive server message
-450	Unable to send client message
-429	Invalid number of input/output parameters for Stored Procedure
-428	Stored Procedure Not Found
-427	Invalid Stored Procedure Name
-426	Error preparing Stored Procedure
-425	Error processing Stored Procedure request

Table 4: The SQLCODE Values for F-15

SQLCODE	Definition
-422	SELECT request processed via 'osql3' cannot contain an INTO clause
-421	Warning: UPDATE or DELETE statement does not contain a WHERE clause
-410	Invalid Directory
-409	Invalid server function
-408	Unable to start server
-407	Unable to Write to Server Master
-406	Unable to Write to Server
-405	Unable to read from communication device
-401	Fatal Connection error
-400	Fatal error occurred
-347	Transaction failed to COMMIT
-346	Transaction failed to COMMIT. Implicit ROLLBACK also failed.
-345	Invalid condition number.
-344	Invalid Transaction Sate - Active SQL-Transaction.
-343	Not in a transaction.
-342	Transaction failed to ROLLBACK.
-341	Transaction failed to COMMIT. Implicit ROLLBACK performed.
-340	Transaction failed to start.
-333	No such index defined
-324	Index with this name already defined for this table
-319	Referenced table has no primary key defined
-315	Constraint or Key not found
-314	Foreign key references non-unique key/column collection
-310	Foreign key references non-existent table
-307	Primary key already defined for this table
-306	Column with this name already exists
-300	DDL not allowed on this table definition
-201	Table or View name not unique
-118	Unknown or non-unique User or Role
-115	Cannot INSERT/UPDATE/DELETE on a read only table

Table 4: The SQLCODE Values for F-15

SQLCODE	Definition
-114	One or more matching rows is locked by another user
-113	%THRESHOLD Violation
-112	Access violation
-111	Cannot INSERT into a 'Default Only' RowID or RowID based on field
-110	Locking conflict in filing
-109	Cannot find the row designated for UPDATE
-108	Required field missing; INSERT or UPDATE not allowed
-107	Cannot UPDATE RowID or RowID based on Fields
-106	Row to DELETE not found
-105	Field validation failed in UPDATE
-104	Field validation failed in INSERT
-103	Positioned UPDATE or DELETE attempted, but the cursor is not positioned on any row
-102	Operation (FETCH/CLOSE/UPDATE/DELETE/...) attempted on an unopened cursor
-101	Attempt to open a cursor that is already open
-99	Privilege Violation
-97	Duplicate <select-list> names found.
-96	Specified <level of isolation> is not supported.
-95	Operation Disallowed by Operation Table
-94	Multi-Conditional OUTER JOINs are not supported.
-93	<number of conditions> for the DIAGNOSTICS SIZE must be exact numeric.
-92	<level of isolation> cannot be READ UNCOMMITTED if READ WRITE specified
-91	<transaction mode> cannot be specified more than once.
-90	Invalid view name
-89	Invalid index name
-88	Invalid field name
-87	Invalid table name
-86	Invalid field definition, no datatype defined
-85	Multiple table %ROUTINE definitions found

Table 4: The SQLCODE Values for F-15

SQLCODE	Definition
-84	Multiple table %NUMROWS definitions found
-83	Multiple table %FILE definitions found
-82	Multiple table %DESCRIPTION definitions found
-81	Column Constraint expected
-80	Integer expected
-79	Referencing key and referenced key must be the same size
-78	Invalid transaction state.
-77	Qualified column reference not allowed in this JOIN context.
-76	Cardinality mismatch between the SELECT-list and INTO-list.
-75	<trim spec> and/or <trim char> required before FROM in TRIM function.
-74	Duplicate <select-list> alias names found.
-73	Aggregates not supported in ORDER BY clause.
-72	Multi-Line Field not valid in ORDER BY clause.
-71	Multi-Line Field must be the Left operand of the Comparison.
-70	Multi-Line Field only valid for LIKE, Contains ([), or NULL Comparison.
-69	SET <field> = <value expression> not allowed with WHERE CURRENT OF <cursor>
-68	DISTINCT not valid in subquery of FROM clause
-67	Aggregate not valid in subquery of FROM clause
-66	Redundant Fields Found in SELECT list
-65	Positive integer constant or variable expected
-64	Incompatible SELECT list is used in INSERT
-63	Data Exception - invalid escape character
-62	Additional new values expected for INSERT/UPDATE
-61	Cursor not updatable
-60	An action (%ALTER, SELECT, UPDATE, etc.) expected
-59	Cannot have more than one field
-58	Object type not found
-56	Action not applicable to an object of this type
-55	Invalid GRANT <role> TO or REVOKE <role> FROM
-54	Array designator (last subscript omitted) expected after VALUES

Table 4: The SQLCODE Values for F-15

SQLCODE	Definition
-53	Constant or variable expected as new value
-52	Cursor (Already/Was Not) DECLAREd
-51	SQL statement expected
-42	Closing quotes ("") missing following pattern match
-41	An extrinsic function call must have the form '$$tag^routine(...)'
-39	No RowID field for table
-38	No master map for table
-37	Contradictory conditions: constants should satisfy condition
-36	Contradictory conditions: 'f IS NULL' vs. 'f=expression'
-35	Contradictory conditions: 'f IS NULL' vs. 'f in range'
-34	Contradictory conditions: 'f IS NULL' vs. 'f=constant'
-33	No field(s) found for table
-32	Outer-Join symbol (=*) must be between two fields
-31	Field not (Found/Mapped/Unique) in Table(s)
-30	Table or View not found
-29	Field ambiguous/not found
-28	Host variable name must begin with either % or a letter
-27	Ambiguous labels for field
-26	Missing FROM clause
-25	Input encountered after end of query
-24	Table or View not found
-23	Label is not listed in FROM
-22	ORDER must specify column names, not numbers, when after 'SELECT *'
-20	Name conflict in the FROM list over label
-19	An aggregate function cannot be used in a WHERE clause
-18	IS (or IS NOT) NULL predicate can be applied only to a field
-17	A for-condition expected after the (in the for-expression
-16	A qualifier SOME or ALL expected after the FOR in the for-expression
-15	A condition expected after NOT
-14	A comparison operator is required here
-13	An expression other than a subquery expected here

Table 4: The SQLCODE Values for F-15

SQLCODE	Definition
-12	A term expected, beginning with one of the following: identifier, constant, aggregate, $$, :, (, +, -, %ALPHAUP, %UPPER, or %EXACT
-11	A scalar expression expected, not a condition
-10	The SELECT list of the subquery must have exactly one item
-9	Incompatible SELECT lists used in UNION
-7	ORDER column is not in the SELECT list
-6	ORDER must specify column number, not names, when after UNION
-5	Column number specified in ORDER does not match SELECT list
-4	A term expected, beginning with one of the following: identifier, constant, aggregate, %ALPHAUP, %UPPER, %EXACT, $$, :, +, -, (, NOT, EXISTS, or FOR
-3	Closing quote (") missing
-2	Exponent digits missing after ’E’
-1	Invalid SQL statement
0	Successful Completion
100	No (more) data

DDL

- Open M with SQL version F.15 supports the additional DDL syntax described in the table below:

DDL Command Syntax	Notes
Alter Table <table name> Drop Primary Key Alter Table <table name> Drop Foreign Key <key name>	Delete a foreign and/or primary key from a table.
Alter Table <table name> Add Foreign Key <key name> (REFERENCING_FIELD) References <table name> (REFERENCED_FIELD) On Update Cascade	Add a foreign key to a table.
Alter Table <table name> Add Primary Key (MY_TABLE_FIELD)	Add a primary key to an existing table definition.

- DDL now supports TINYINIT datatypes as synonymous with SMALLINIT and DATETIME as synonymous with TIMESTAMP.
- DDL Options have been added to the SQL / Base Table Options menu on the System Configuration screen. These options allow you to select alternatives to returning SQL errors and automatically invoke the Hide From Select * feature for certain DDL statements.

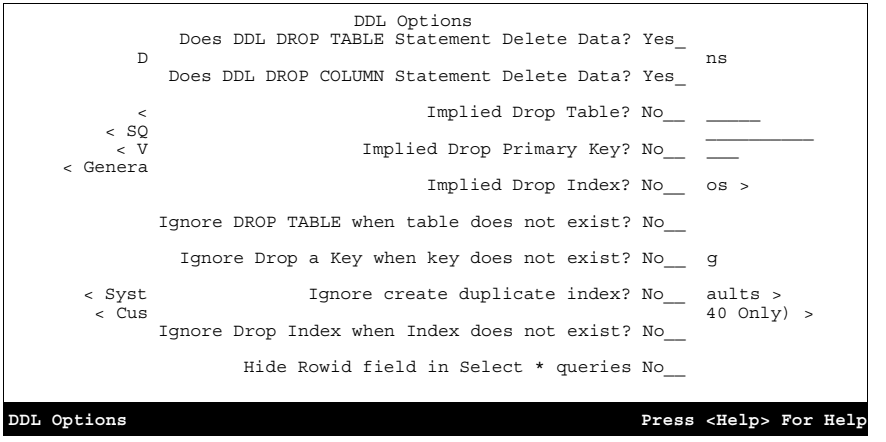

Table 5: System Configuration Menu — DDL Options

DDL Option	Description
Does DDL DROP TABLE Statement Delete Data?	This is a system wide default which can be overridden in the DDL statement by using the %NODELDATA and %DELDATA keywords. The setting is checked at DDL statement compilation time. Enter ***Yes*** to delete the table's data with the drop table command. Enter ***No*** to retain the table's data.
Does DDL DROP COLUMN Statement Delete Data?	Enter ***Yes*** to delete the column's data with the drop column command. InterSystems suggest specifying ***Yes*** here to prevent orphaned data from wasting disk space. Enter ***No*** to retain the data in the global(s) after the column is dropped'. You may need to enter ***No*** if your application references the data directly through Open M.
Implied Drop Table?	Enter ***Yes*** to perform an implied drop table if a create table statement is executed for a table that already exists. Enter ***No*** to return SQLCODE -201.
Implied Drop Primary Key?	Enter ***Yes*** to perform an implied drop primary key if an attempt is made to create a primary key for a table that already has one. Enter ***No*** to return SQLCODE -307.
Implied Drop Index?	Enter ***Yes*** to perform an implied drop index if a create index statement is executed for an index that already exists. Enter ***No*** to return SQLCODE -324.
Ignore DROP TABLE when table does not exist?	Enter ***Yes*** to ignore drop table statements on tables that do not exist. Enter ***No*** to return SQLCODE -30.
Ignore Drop a Key when key does not exist?	Enter ***Yes*** to ignore drop key statements when the table does not have a key of that name defined. Enter ***No*** to return SQLCODE -32.
Ignore create duplicate index?	Enter ***Yes*** to ignore a create index statement when there is already an existing index on the same columns. Enter ***No*** to return SQLCODE -325

Table 5: System Configuration Menu — DDL Options (Continued)

DDL Option	Description
Ignore Drop Index when Index does not exist?	Enter ***Yes*** to ignore drop index statements when the index being dropped does not exist. Enter ***No*** to return SQLCODE -333.
Hide Rowid field in Select * queries	Enter ***Yes*** to exclude the table rowid field from SELECT * queries for tables created using DDL. Enter ***No*** to include the rowid in any Select * queries.

- DDL now uses 10000 as the default number of rows in a base table. The previous default value was 250.
- Open M with SQL version F.15 provides the ability to turn transaction processing off within DDL statements by setting the local variable, %noTP, to the value 69.
- The default length for LONG VARCHAR is set to $$$MaxCharLength if you do not specify a length in the DDL CREATE statement. For most systems, $$$MaxCharlength is 32767.
- SQL SET OPTION statements and UPDATE STATISTICS statements no longer cause compilation errors.
- When unique indexes are created through DDL, the execution of the statement adds table validation code for insert and update to check the uniqueness of that field or combination of fields in that index. If that field or combination of fields exists, the statement returns SQLCODE -107.

ODBC

ODBC queries via ODBC using ODBC escape syntax for dates that are against table with conditional indexes on the data field will use that index.

Forms

- When using popup menus, if the screen is cleared before calling a second popup, Caché displays the title of the second popup menu.
- Timed reads have been changed to timed hangs after redisplaying horizontal menus in forms. This was done to avoid situations where characters could seem to disappear if the user was typing ahead.

Utilities

- The new %uredit routine allows you to edit any .MAC, .INC, or .INT routine with the text editor of your choice. For example:

```
d ^%uredit("vi")
d ^%uredit("emacs")
d ^%uredit("notepad")
d ^%uredit("wordpad")
```

  The %uredit routine prompts you for a routine name, and puts you in your chosen editor.

Note: The location of the editor should be in your path

Import/Export

- When using Copy Object Export / Import, we support importing from files created with versions prior to F.15. After importing from older versions, you must run all the conversion tasks in the import namespace from the version the export files were created with and up. After running any conversion tasks, the imported objects must be re-compiled. You may not import from export files created with F.15 into a pre-F.15 system.
- General Export and Import have been enhanced to include M / Pact report triggers.
- There is a new field, Open Parameter, on the Import Run screen which allows you to specify the open parameters for a general import.

Data Dictionary

- An enhancement has been made to default structure mapping where the original nodes and pieces of the map data definition are retained when the map is recreated.

  Previously, if a field was deleted, the nodes and pieces would be shuffled down, causing field_b being mapped to where field_a used to be.
- The map name for default structure tables is now the same as the index name.

Terminal Characteristics

- When you run Open M with SQL version F.15 from Caché, the system uses the ^%IS global to determine the default terminal type displayed.

Corrections

SQL

- InterSystems has corrected a problem with false cardinality (SQLCODE -76) errors in override lookups used by forms when the override query contains DISTINCT BY(field1, field2, ...fieldn). This error occurred in Open M with SQL versions from F.9.
- InterSystems corrected where default structure tables with index maps would not retain the original mapping locations of fields after a field had been deleted from the table.
- InterSystems corrected a problem in which an SQL-Standard query with aggregate functions in the SELECT-list and no GROUP BY clause, would return exactly one row, as specified by the SQL Standard, even in the case where the aggregate functions are applied to a result table with zero rows.
- InterSystems fixed a problem with the SQL code generator. Selecting long variable names in a map with a large number of subscripts could confuse the SQL generator and cause it to generate incorrect code.
- InterSystems fixed a problem with queries using order by or group by. The would only return correct results if the fields referenced in the group or order by clause were text data type and the result set had rows with null values for these fields.
- 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.
- Open M with SQL supports the following syntax:
  `INSERT ... ( <query expression> )`
  In the past, the <query expression> would give an error if there were parenthesis around it.
- InterSystems has corrected a problem with General Views containing <sub-queries> within the SELECT statement. This caused problems when an outer query used the General View as one of the tables in the FROM clause and tried to SELECT the field that was the <sub-query> within the General View.
- InterSystems has corrected a problem with General Views where column qualification within the view as not being processed correctly.
- InterSystems has corrected a problem where parameter information was not returned to the Client when the parameter was part of a Subquery.
- The SUBSTRING function now correctly returns the number of characters specified in the function.

For example:

```
SUBSTRING('Another Test' FROM 2 FOR 4)
```
returns: noth

- InterSystems has corrected a problem where queries with order by or group by clauses could give incorrect results if the fields referenced in the clause were text data type and the result set had rows with null values for these fields.
- InterSystems has corrected a problem where generated queries against tables with override full row references ignored the override reference in piece maps.
- InterSystems fixed a problem with multiple LIKE (and %STARTSWITH) conditions in a WHERE clause when more than one of the like conditions where on fields looped on as subscripts (usually in an index map).
- InterSystems corrected a problem where not all index maps for a table would get recreated after dropping an index.

DDL

- InterSystems corrected an <UNDEFINED> error while using DDL to create tables.
- InterSystems corrected an <UNDEFINED> error while dropping index using DDL.
- DDL now recognizes underscores in referenced field names for FOREIGN KEYS.
- DDL export table now correctly recognizes the starting table. The DDL export was not recognizing the first table in the view which caused the DDL export to come out wrong and could be not be imported back in.

ODBC

InterSystems fixed a problem where using UPDATE or DELETE could cause all of the ODBC cursor information to be deleted. This resulted in problems accessing information needed if multiple sets of host-vars were received for the same statement, or if the same “prepared” statement was executed again.

Client/Server

- FileMan dates before the year 1840 are now returned properly to the client from the Caché SQL Server.
- InterSystems corrected a problem in the Relational Server where a DDL statement would fail to compile if any of the identifier names began with x###.
- InterSystems fixed a problem with %qcfcr during keyboard generation that produced and undefined error.
- A number of changes were made to the way F handles dates to allow better client access from FileMan.
- Changed the entries from %msqlapi to be recognized in %qcrsapi.

Unicode

- InterSystems has corrected a problem where identifier (Table, Field, View, etc.) names with wide characters could not be created through DDL.
- InterSystems corrected a problem with concatenating strings in unicode.

Import/Export

- InterSystems has corrected a <STORE> error importing tables or forms with 90+ lookups defined.
- InterSystems has corrected a problem in the export utility where variable placement information for menu objects were not being exported properly.

M/PACT

- InterSystems has corrected a problem that stopped a chain report from running to a file or printer if one of the middle reports had no data.

Forms

- InterSystems fixed a problem with forms where you could get a PARAMETER error when doing a designative display lookup.
- InterSystems corrected a problem with key macros in “new style” forms where the keys were not being used correctly.
- InterSystems corrected a problem where the cursor displayed on the wrong line of a multi-row form.
- InterSystems corrected a problem in a multi row forms with pre-field goto next field triggers on the last field of the row where the cursor would not move off the <proceed> on up arrow has been corrected.
- InterSystems corrected a problem in which multiline fields with no rows would sometimes get reported to the ODBC client as having the non-NULL value of 0.
- InterSystems corrected a problem where if the Cache' system was defined to add the nodename to the PID for the $J value, the CHUI full screen editor would not work.
- The Form List report generator now produces reports that include form only forms.
- InterSystems corrected a problem in the form generator that caused a MAXSTRING error while compiling if you had a very large number of fields on the form.
- InterSystems has corrected a problem that caused a parameter error if you called a variable placement window with caller ID with a window trigger.
- InterSystems fixed a problem with designative display fields on multi row forms. If the Designated Reference has override lookups and the user does a partial lookup, the user will get a message saying values have changed.
- InterSystems corrected a problem with the form generator. Now calls to a form in insert mode where you pass a default value array will not pass those values along to other forms called by the current form.
- InterSystems has corrected a problem in multi-row forms. If you went in and out of a multi row form when the cursor was on the empty row at the end of the form, the system would misinterpret the correct number of row. Potentially, an up arrow could cause field data to disappear.
- A problem has been corrected which limited the number of windows on a form to 25. Forms with more than 25 windows now compiles without getting <LABELREDEF> errors.
- InterSystems has corrected a problem with designative display fields on multi row forms. If the designative reference has override lookups

and the user does a partial lookup they get a message saying values have changed.

- InterSystems has corrected a problem in the form generator. If you call a form in insert mode and pass a default value array, the values are not passed onto other forms called by this form.
- A correction has been to properly use key macros in new style forms.
- InterSystems has corrected a problem where a multi row form that had a pre-field goto next field trigger on the last field of the row the cursor would not move off the <PROCEED> on up arrow.
- InterSystems has changed a timed read to a timed hang after redisplaying horizontal menus in forms. This was done to avoid situations where characters could seem to disappear if the user was typing ahead.
- A problem has been corrected where a multi row form did not compile correctly when a window contained a rowid field that was output only.

Utilities

- InterSystems has corrected an <UNDEFINED> error in the Object Integrity Checker where attempting to fix Filed “ifn” index problem would fail.
- InterSystems has corrected a <NOLINE> error running the Conversion Manager / F* to F.10 / Reserve Word / Identifier Conflict option.
- InterSystems corrected a problem with genkeys^%qcfcr (formerly genkeys^cafcr) where running the utility without changing to the manager’s directory would result in a namespace error.

Base Tables

- InterSystems has corrected a String Too Long / <MAXSTRING> error caused by numerous multiple-choice fields with very long field names.
- InterSystems has corrected a problem where default structure tables with index maps did not retain the original mapping locations of fields after a field had been deleted from the table.
- InterSystems has corrected a problem where not all index maps for a table would get recreated after dropping an index has been corrected.

Available Documentation

Open M with SQL

The documentation set for InterSystems Open M with SQL relational database product includes the following manuals:

- This *Open M with SQL — Version F.15 Release Information*
- *Open M with SQL — Version F.14 Release Information*; Revision Date: July, 1997.
- *Caché SQL - Version F.13 Release Information;* Revision Date: December, 1997.
- *Caché SQL - Version F.12 Release Information;* Revision Date: September, 1997.
- *Caché ODBC Driver Guide* —Version 3.1; Revision Date: April, 1999.
- *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.

ADDENDUM

A

Converting Your Applications to the New Global Structure for Routines

New Global Structure for Routines

Open M with SQL routine code is now stored in five separate globals. Previously, all routines were stored exclusively in the ^ROUTINE global[1]. This was done to distribute routine source code across directories and/or systems. When you upgrade to Open M with SQL Version F.15 from version F.10 or earlier, you must convert your existing Open M with SQL routines to the new global structure.

Open M routines can have different extensions and version numbers. The extension describes the type of routine file, and the version differentiates between multiple copies of the same routine. A routine can have the following extensions:

- .MAC — Macro Source Routine
- .INT — Intermediate Source Routine
- .INC — Include File
- .OBJ — Compiled Object Code

1. For Open M with SQL Version F.10, the global routines were called ^ROUTINE, ^mINC, ^mINCSAVE, ^mMAC, and ^mMACSAVE.

Under the new global structure for routine storage, the intermediate code can be in one directory, the macro source code in another and etc. The five globals used for Open M with SQL routine storage are:

- ^ROUTINE
- ^rINC
- ^rINCSAVE
- ^rMAC
- ^rMACSAVE

Routine Storage Format

Table A-1 shows how routines are mapped to the new structure:

Table A-1: Routine Storage Format

Routine Extension	Old Global Format	New Global Format
.MAC	^ROUTINE(0,"MAC",*Version, Routine_Name*,0)=*Date of last edit* 0,0)=*Number of code lines* 0,n)=*Line of code*	^rMAC(*Routine_Name,0*)= *Date of last edit* *0*,0)=*Number of code lines* *0*,n)=*Line of code*
		for backup versions[1]: ^rMACSAVE(*Routine_Name,Version*)= *Date of last edit* *Version*,0)=*Number of code lines* *Version*,n)=*Line of code*
.INC	^ROUTINE(0,"INC"*,Version*, *Routine_Name*)	^rINC(*Routine_Name,0*)= *Date of last edit* *0*,0)=*Number of code lines* *0*,n)=*Line of code*
		for backup versions: ^rINCSAVE(*Routine_Name,Version*)= *Date of last edit* *Version*,0)=*Number of code lines* *Version*,n)=*Line of code*
.INT	^ROUTINE(*Routine_Name*,0)= *Date of last edit* 0,0)=*Number of code lines* 0,n)=*Line of code*	Same as old format

1. The highest version number reflects the most recent backup.

Parameter	Meaning
Date of last edit	Date that the routine was last edited. Stored in the M standard $H format.
Number of code lines	The total number of code lines in the routine.
Lines of code	The individual code and comment lines that make up the routine.
Version	The version number of the routine. Zero (0) is the current version. Highest version number is the latest backup.
Routine_Name	Name of the routine stored in this global node

ROUTINE Global Conversion Utility

When you upgrade to Open M with SQL Version F.15 from version F.10 or earlier, you must convert your existing Open M with SQL routines to the new global structure. Follow the appropriate procedure for your system as described below:

For Open M NextGen 2.0 Systems:

Any existing routine global names are converted as part of the system conversion routine, %SYSCONV. See your *Open M NextGen Installation Guide* for information on system conversion.

For Open M with SQL 6.x, Open M with SQL for DSM, and Open M with SQL for DTM Systems:

The %urconv utility converts the ^ROUTINE global to the new global storage format. To convert your existing routines to the new global structure:

```
>d all^%urconv
```

Note: The %urconv utility must be run on a per directory basis for **all** application directories. It should not be run in the manager's directory or the Open M with SQL common directory.

The entry points to the ^%urconv routine are listed and described in the table below.

Table A-2: Entry Points into %urconv

Entry Point	Action
d all^%urconv	Converts all .MAC and .INC files in the current directory to the new structure.
d ^%urconv	Converts selected routine(s) to the new structure. Enter routine specifications or ? at the Routine prompt.
d rtnset^%urconv	Converts your predefined routine sets to be compatible with the new format. These routine sets are stored in the global ^mutil("rset").

Setting the Number of Backup Versions

You can set the maximum number of versions maintained for Macro Source routines (.MAC) and Include files (.INC). Intermediate source routines (.INT) can only have one copy at a time. Run the %urverma routine to set this version limit:

```
>d ^%urverma
Number of versions to keep for .MAC:  4 =>
Number of versions to keep for .INC:  4 =>
```

Note: The version limit should be a number greater than 0.