



# What's in OpenQM but not in U2?

## Connection

- Port mapping - links telnet port number to QM user number
- Use of QM as an element of a Linux pipe
- Serial port connection

## File System, Dictionaries and Conversions

- Optional case insensitive record ids on a per-file basis
- Optional case insensitive alternate key indices
- Integrated QMNet as a standard component
- Virtual File System for access to alien file systems
- Extended Q-pointers for QMNet access to remote servers
- Deferred dynamic file split/merge
- Dictionaries - L-type records to minimise the need for TRANS() I-types and T conversions
- Compiled A/S-types for improved performance
- C-type records for complex dictionary calculations
- Nested compound I-types
- TRANS() can retrieve a calculated value
- Additional conversion codes - Base64, Boolean, ISO dates, ANSI X12 dates, Integer, Field extraction
- Extended triggers - post read, pre/post write, pre/post delete, pre/post clear
- Ad hoc encryption
- Epoch based date/time for multi-time zone applications

## Command Environment

- Cursor key driven command editor
- Personal vocabulary (per user or shared by groups)
- Simpler use of security subroutines on V-type VOC entries
- Largely case insensitive
- ON.LOGTO VOC item, executed by LOGTO before moving to new account
- Extended file syntaxes for remote files and access by pathname
- Dynamic menus allow run-time control of which options are displayed
- Extended inline prompts - substitution of select list items, @-variables, environment variables, function values, SYSTEM() values, ranges of command line tokens
- Pick style form queue emulation
- Print prefix and overlays for forms printing
- Hold file records locked while being written
- Optional call to user written subroutine to log hold file record creation

- PCL printer graphics
- ALIAS to remap command names
- CREATE.FILE option to create in non-default location
- Simple display of internal format date and time
- DISPLAY supports cursor management, reducing need to write programs for simple screens
- User defined @-variables allow programming style operations in paragraphs
- LOGMSG to add messages to the system error log file
- Immediate messaging via pop-up windows
- Run a VOC style item from alternative files
- Enhanced UPDATE.RECORD with interactive mode
- Suspend/resume COMO file creation
- Option to run phantom processes without logging

## Query Processing

- Enhanced heading/footer capabilities
- Report styles for colour or font highlighting
- \$QUERY.DEFAULTS to set standard query default actions
- Use of dictionary links to access remote files without needing TRANS() I-types or T conversions
- Full or partial pan of reports wider than the screen
- Scroll back through report pages
- SHOW command for interactive select list generation
- ABSENT.NULL option to handle mis-matched records in report built from data in multiple files
- BETWEEN operator for inclusive ranges
- BOXED option for enhanced report appearance
- Breakpoints - Multiple B elements to capture heading data
- Extended column heading format controls
- CSV and delimited reports
- CUMULATIVE option for running totals
- Case insensitive selection clause options - not just mapped to uppercase
- IN and NOT.IN selection operators to compare field value with a stored list
- Extended page heading/footer control options
- Optional file locking during report for data consistency
- Overlay pages for forms printing
- Page sequence control for merged reports
- REPEATING to repeat single valued item within multivalued data
- SAVING clause can handle multivalued data

## QMBasic

Enhanced heading/footer capabilities  
Simple mark translation suppression on directory files  
Set exit status for operating system error handling  
Extended sequential file processing statements ( CSV handling, append, overwrite)  
Object Oriented Programming  
Automatic cataloguing option after successful compilation  
Variable length argument lists to subroutines and functions  
Terminfo based function key recognition (BINDKEY, KEYCODE, INPUTFIELD)  
Local functions within source program modules  
Arguments to internal subroutines  
Private data in internal subroutines/functions, fully stacked for recursive calls  
Save/restore part or all of screen image  
Ability to open a file for non-transactional update  
User defined break handlers for application level quit key handling  
CHILD() - Check if child phantom process is still running  
CLEAR.COMMON - Extended to allow clearing of specific named common  
CROP() - Remove redundant mark characters  
IDIV() - Integer division - also as // operator  
INPUTCSV - CSV format input parsing  
INPUTFIELD - Extended INPUT with function key handling  
LISTINDEX() - Find item in arbitrarily delimited list  
LOCATE() - Function call style LOCATE, very useful in I-types  
LOGMSG - Add message to system error log  
MAX(), MIN() - Find maximum or minimum of two values  
MD5() - Create MD5 message digest value for a given string  
OSREAD, OSWRITE, OSDELETE - Simple operating system file read, write, delete  
OUTERJOIN() - Simple interface to indices for use in I-types  
PAUSE / WAKE - Synchronise with some other process  
REMOVEF() - High performance extraction of delimited items  
SHIFT() - Logical shift  
SUBSTITUTE() - Multivalued text substitution  
TESTLOCK() - Test state of task lock  
VSLICE() - Extract a row of a multivalued data set  
QMClient security enhancements complete access control  
Development Aids

Paragraph tracing  
S (stack), W (where) and P (process dump) options on quit key actions  
DELETE.COMMON, LIST.COMMON  
Real time file statistics monitoring  
Equate token generation to ensure consistency between dictionaries and programs  
Hot Spot Monitor to identify performance bottlenecks  
LIST.FILES to determine names of all open files and who has them open  
Full screen debugger over telnet connection  
Debug phantom and QMClient processes  
Process dump files - Complete state of an application at a run time error or on demand  
Windows code page selection for non-default character sets  
Simple screen builder for rapid application development  
Full screen editor

### Questions?

Contact our sales team at [openqm@zumasys.com](mailto:openqm@zumasys.com) or visit [www.openqm-zumasys.com](http://www.openqm-zumasys.com)



## About Zumasys

Zumasys helps companies of every size transition their infrastructure and applications to the cloud. With Zumasys cloud services, customers can easily access the latest software and hardware technologies over the Web, allowing them to focus on growing their core business instead of managing their IT infrastructures. Zumasys delivers personalized service, integrated disaster recovery and the confidence companies need to outsource the hosting of all their applications, including legacy MultiValue systems.

### IRVINE HEADQUARTERS

9245 Research Drive  
Irvine, CA 92618  
866-ZUMASYS (986-2797) | phone  
949-334-0287 | local  
949-334-0288 | fax

### LAS VEGAS OFFICE

6405 South Tenaya Way, Suite 110  
Las Vegas, NV 89113  
866-ZUMASYS x328 | phone  
702-333-2194 | local  
949-334-0288 | fax

### UK HEADQUARTERS

Hamilton House  
111 Marlowes  
Hemel Hempstead Herts HP1 1BB  
United Kingdom  
0808 189 3266 | phone  
020 3514 8368 | local

For a complete list of locations, visit [www.zumasys.com/contact](http://www.zumasys.com/contact)

[www.zumasys.com](http://www.zumasys.com)

© 2016 Zumasys, Inc. All Rights Reserved. DS-041516



All product and service marks contained herein that are not Zumasys, Inc. proprietary marks are the trademarks of their respective owners.