RELEASE INFORMATION

Northgate Reality

Release Information

Version 15.0

Date: 05 December 2012

Northgate Information Solutions UK Limited
Peoplebuilding 2
Peoplebuilding Estate
Maylands Avenue
Hemel Hempstead
Herts
HP2 4NW

Tel: +44 (0)1442 232424
Website: http://www.northgate-reality.com/

Contents

Introduction

Reality is a software environment that supports multiple MultiValue SQL-enabled databases on a single host and includes a range of powerful utilities for building, managing and accessing the databases.

The release information in this document applies to Reality V15.0 for UNIX and Windows. Reality V15.0 adds new features and enhanced compatibility with similar database systems. Faults reported since the production release of Reality V14.0 have been resolved. See *New Features in Reality V15.0 (page 12)* and *Fault Resolutions (page 29)* for more details.

Reality V15.0 is supplied as a downloadable ISO image file or on a DVD. These both contain:

- The Reality database software.
- User documentation.
- UNIX-Connect - networking software that provides communications between Reality databases and between Reality and host system environments.
- Reality Remote Tape - server software that allows a Reality host to use tape units on remote systems.
- PCSNI - client software that allows communication between a PC and a Reality database.
- JReal - client software that provides the Java programmer with the ability to run Remote Basic subroutines and to write custom servlets to access a Reality database via RealWeb.
- RealWeb - software that provides a web developer with DataBasic experience with access to data held in a Reality database.
- RealSQL-JDBC Driver - client software that provides a standard API for Java applications, applets and servlets using SQL to access data.
- RealSQL-ODBC Driver - client software that allows PC applications to access data using SQL.
- RealEdit - a Reality editor that runs on Windows PCs.
- Remote Basic ActiveX Control - a DLL that can be used in PC programs written in Visual Basic to run Remote Basic subroutines on a Reality database.
- Reality and RealWeb demonstration software.

A second ISO image file/DVD contains the Web Services feature (including the Jetty web server). This can be downloaded from the Reality website by selecting: **Support** > **Downloads** > **System Components**.

Retirement Notice

This version of the software supersedes all previously released versions. Northgate policy is to withdraw support for previous versions six months after a new release. The relevant date for this software can be obtained from your Northgate representative or the Reality website.

Backwards Compatibility

Northgate attempts to make each new version of Reality fully backwards compatible with previous versions. However, fault resolutions and new features can, in some cases, result in changes to menus and prompts displayed by host and TCL utilities. Scripts which automate such utilities may therefore need to be reworked after upgrading or installing updates.

Deliverables

All of the software comprising this release is supplied on two downloadable ISO image files or two DVDs, with electronic versions of all documents (including this one).

Reality ISO image file/DVD

- PDS History Tool
- Reality
- User Documentation
- UNIX-Connect
- Reality Remote Tape
- PCSNI (client)
- JReal (client)
- RealSQL-JDBC Driver (client)
- RealSQL-ODBC Driver (Windows client)
- RealSQL-ODBC Driver (UNIX client)
- RealWeb HTML
- RealWeb Servlets
- Reality Demonstration
- RealWeb Demonstration
- WinSNI Configuration Editor (client)
- RealEdit (client)
- Remote Basic ActiveX Control
- TCP Bridge

Web Services ISO image file/DVD

- Reality Web Services

Reality Website

Visit the Reality website at www.northgate-is.com/reality for:

- Product information (select Products).
- Downloads (select Support > Downloads).

- The latest documentation (select Products > On-line Reality Documentation).

  Reality is supplied with comprehensive on-line documentation for viewing in a web browser. Refer to the Document Directory in the on-line documentation for details. Note that the documentation is subject to change and it is essential that you have the latest version. You should regularly download this from the Reality website.

  We are constantly trying to improve the Reality documentation, so please send us your comments. Every topic in the documentation includes a link to a comment form.

- Enquiries (select **Contact us** > **Enquiries**).

Prerequisites

Reality on UNIX

- One of the following:

  SUN SPARC running Solaris 11 or 10 (64-bit only).

  IBM pSeries (RS 6000), running AIX 7 or 6 (64-bit only).

  PC with Intel Pentium processor or equivalent running one of the following versions of Linux: Red Hat version 9, ES/AS versions 2.1 or 3 (32-bit), Red Hat ES 6 or 5 commercial release (64-bit), CentOS 6 or 5 'open' software (64-bit).

- 128Mb RAM minimum (512Mb recommended), plus 2-6Mb per Reality User. See also *Memory (page 10)*.
- 1 GB of available space to accommodate setup (actual hard disk used once installed will be between 220Mb and about 350Mb, depending on the system components installed, plus space for databases).
- Korn shell.
- Perl - this is normally supplied with the operating system.
- UNIX-Connect for networking (supplied in the Reality ISO image file or DVD).
- Northgate Customisation (a CD is available for SUN - for other operating systems, contact Northgate).
- C compiler (on Solaris, if a C compiler is not available you can install the GNU C Compiler from the Northgate Customisation CD).

UNIX-Connect

'rosi' UNIX user id with a home directory on a file system with at least 25 Mbytes free.

Reality on Windows

- PC with Intel x86-64 processor or equivalent, 200MHz or faster, running Windows 8, 7 or XP, Server 2012, 2008 or 2003 (32-bit or 64-bit).

  Note: Reality is not available for Windows NT, 95, 98, 98 SE, ME or Vista.

- 64Mb RAM minimum (128Mb recommended), plus 2-6Mb per Reality User. See also *Memory (page 10)*.
- 500Mb of available disk space to accommodate setup (actual hard disk used once installed will be between 150Mb and about 350Mb, depending on the system components installed, plus space for databases).

  It is recommended that Reality is installed on an NTFS partition.

- The Reality database can be loaded on to a Primary Domain Controller, Backup Domain Controller, stand-alone member server or WorkStation.

- Northgate can take no responsibility for the processor and memory requirements of other applications running on a Windows server. Ideally, Reality should be loaded on a dedicated server.
- Reality backup and restore is supported on 4mm, 8mm and DLT tape units.
- Using at least a dual processor system is highly recommended.

Memory

Reality memory usage is difficult to predict, but as a rough sizing guide use 128Mb for the system and then 2Mb to 6Mb per user, depending on type of user and application. Performance problems are generally caused by lack of memory. If the server is not dedicated to Reality then other application memory requirements must be added to this.

Foreign Database Support and SQL View

These features require a working ODBC installation, with appropriate ODBC driver(s), on the Reality system.

External Components

PCSNI

- PC with Intel Pentium processor or equivalent, 200MHz or faster, running Windows 7 or 8, XP (Home or Professional), Server 2003 (Professional or Server), Server 2008 (32-bit or 64-bit).
- A LAN card.
- TCP/IP networking.
- A way of resolving host name and IP addresses (for example, a hosts file or a domain name service).

RealEdit

- PC with Intel Pentium processor or equivalent, 200MHz or faster, running Windows 7 or 8, XP (Home or Professional), Server 2003 (Professional or Server), Server 2008 (32-bit or 64-bit).
- Northgate PCSNI software V2.3.1 Rev C or later.

JReal

Java Run Time Environment (JRE) version 1.4.1_02 or above. If this is not available, it can be downloaded from Sun's Java web site (http://java.sun.com/j2se/).

RealSQL-ODBC Driver

- PC with Intel Pentium processor or equivalent, 200MHz or faster, running Windows 7 or 8, XP (Home or Professional), Server 2003 (Professional or Server), Server 2008 (32-bit or 64-bit).
- Northgate PCSNI software V2.3.1 Rev C or later.
- Any ODBC Level 1 or 2 compliant application.
- A Winsock compliant TCP/IP transport stack for TCP/IP connections.
- If you are connecting to a Reality database on a UNIX host, the host will need UNIX-Connect.

The PC applications and transport stacks use large amounts of memory. It is therefore essential that PCs running this package are configured for the optimum use of memory; otherwise it is possible that GPFs and other memory type errors will occur.

RealSQL-JDBC Driver

- Java Run Time Environment (JRE) version 1.4.1_02 or above. If this is not available, it can be downloaded from `java.sun.com/j2se`.
- If you are connecting to a Reality database on a UNIX host, the host will need UNIX-Connect.

RealWeb

- A web server with support for Java servlets. (This can be on the same system as the Reality database or on another system.) On web servers that do not support servlets, plug-ins can be used to add servlet support.
- Java Run Time Environment (JRE) version 1.4.1_02 or above. If this is not available, it can be downloaded from `java.sun.com/j2se`.
- If you are connecting to a Reality database on a UNIX host, the host will need UNIX-Connect.

Remote Tape Server

Any UNIX or Windows system that supports Reality.

Reality Web Services

- A web server with support for Java servlets (Jetty is supplied). This can be on the same system as the Reality database or on another system. On web servers that do not support servlets, plug-ins can be used to add servlet support.

  Note: Reality Web Services have been tested on Jetty 5.1.4 and Tomcat 5.0.

- Java Runtime Environment (JRE) or Java Development Kit (JDK), version 1.4.1_02 or later. For Jetty, you can use either; for Tomcat you must use a JDK.

Remote Basic ActiveX Control

- PC with Intel Pentium processor or equivalent, 200MHz or faster, running Windows 7 or 8, XP (Home or Professional), Server 2003 (Professional or Server), Server 2008 (32-bit or 64-bit).
- *PCSNI - client software that allows communication between a PC and a Reality database. (page 6)* software V2.3.1 Rev C or later.

Online Documentation

The online documentation can be installed on a web or file server, or on individual PCs running Windows or Linux. It can also be viewed from the Reality DVD.

The online documentation is compatible with most contemporary web browsers. The following are known to be suitable:

- Internet Explorer 7.0 or later (PC only)
- Mozilla Firefox 3.5 or later (PC or UNIX)
- Google Chrome

New Features in Reality V15.0

Reality V15.0 contains a large number of new features since the release of V14.0, such as full 64-bit support, outgoing web services, updated ODBC support, item time stamps and so on. Other important changes are listed under the broad heading of 'MultiValue'. Compatibility, including many features that are of immediate benefit to all Reality users – like the use of case insensitive searches while selecting and then editing data items. So be sure to review the following list of new features introduced by V15.0.

Some of the new features have come from the user feedback that we receive during the life of a release, so please continue to use the 'Comment on this topic' links at the top of each topic in the Online Documentation, or visit the Reality website at www.northgate-is.com/reality, in order to help us to improve your Reality.

Full 64-bit Support Across all Platforms

Significant changes have been made to deliver optimal performance and extended datum sizes across all platforms. For Microsoft Windows, continued 32-bit compatibility ensures that Reality remains useable on legacy laptops and low end servers.

Outgoing Web Services

Reality's new outgoing web services feature enables DataBasic programmers to call third-party web services over HTTP.

SQL for Reality

Reality SQL/ODBC now complies with ODBC version 2.5 and the ODBC Applications Programming Interface (API) Level 2.

Indexes

- The **ICOUNT** and **ISELECT** TCL commands will now accept symbols as comparison operators, as alternatives to the alphabetic ones previously available.

  In addition, when using the **AND** operator, the higher key value can now precede the lower key value.

- The **SELECT-INDEX** TCL command will now accept multiple key values separated by colons, corresponding to the **BY** clauses in the index definition.
- For indexes that were created from dictionary definitions that include output conversions, the **ICOUNT**, **ISELECT** and **SELECT-INDEX** TCL commands apply those output conversions as input conversions to the literal data specified as appropriate. This behaviour can be inhibited with the EXTERNAL.KEY environment option.
- **POSITION** statement (DataBasic). New keywords allow you to position the read pointer at the start of an index and at the position following an entry that matches a key value.
- **LIMIT.INDEX** statement (DataBasic). It specifies the end of a range of positions within an index and must be used in conjunction with the **POSITION** statement.

- For indexes that were created from dictionary definitions that include output conversions, the **POSITION** and **LIMIT.INDEX** statements apply those output conversions to the literal data specified as appropriate. This behaviour can be changed with the EXTERNAL.KEY compatibility switch.
- Automatic indexing is now available for Remote Q/Pointers, for deployment across multiple systems.
- All of the direct index commands, plus automatic index access through ENGLISH, now work through remote QPTRs.

Item Time Stamp

In the same way that it is possible to determine the date on which an item was last updated, Reality now allows you to determine the time of day when the item was last updated. This time is expressed as the number of milliseconds since midnight, although currently the granularity is 1000 milliseconds (that is, one second).

Time stamps are available for items read from standard Reality files, local or remote, and for items read from directory view files. When the item is remote the remote Reality must be release V15 or later. Non-zero time stamps are immediately available on directory view items which already exist, but are available on items in standard Reality files only when they are updated or created. Time stamps for items read from other types of file, such as (make) special files, are zero.

SYSTEM(111) returns the time when the most recently read item was last updated.

Time stamp values can be used as operands in algebraic (A code) and mathematical (F code) English conversion codes.

File Save and Restore

From V15.0, items that are saved from the local database by using the SAVE verb (or a derivative) include the item time stamp. When these saves are restored to the same release of Reality or later, the time stamps are applied to the restored items.

When a file which has been saved from an older version of Reality is restored, where the items do not include time stamps a time stamp of zero is applied to the restored items.

Items saved using T-DUMP do not include time stamps; T-DUMP save format is dictated by the SMA standard which does not include meta-data such as date and time stamps. Items restored using T-LOAD have a time stamp of zero applied.

Configurable 2-Digit Date Window

Reality has always allowed dates to be entered into the system with a one- or two-digit year, but until now it has always used a fixed date window of 1930-2029 (so that, for example, 11 means 2011 rather than 1911).

With this release, this date window is configurable, either at the database level by means of the **IDateWindow** database configuration parameter, or at the user level by means of the **IDATE-WINDOW** TCL command.

Memory Mapped I/O

Memory mapped I/O has been introduced in Reality for Windows, and may produce performance gains. If necessary it can be suppressed using the new environment variable, REALUSEMMAP.

MultiValue Compatibility

Reality V15.0 has been further enhanced to improve compatibility with other MultiValue systems. In addition, the following features simplify migration to Reality from other MultiValue systems.

- *MultiValue Compatible Databases (page 14)* (see below). This feature allows you to select all the options necessary to emulate a different MultiValue system with a single command.
- *Case-insensitivity (page 14)*. Options are available to make Reality operate in either the default, case-sensitive mode or a D3 compatible, case-insensitive way. Refer to the sections *Data Case-insensitivity (page 14)*, *Keyword Case-insensitivity (page 15)* and *Item-id Case-insensitivity (page 16)*.

  You should not normally need to use these, as the correct options are set when you specify the MultiValue emulation for a database.
- *DataBasic Environment Control (page 17)*. This allows the run-time behaviour of a DataBasic program or subroutine to be determined by the compiler used and the MultiValue compatibility mode.
- *Logon Programs (page 18)*. Reality logon programs have been enhanced in various ways.
- *TCL (page 20)*. The command line language.
- *Other MultiValue Features (page 19)*.

MultiValue Compatible Databases

The **mkdbase** host command has an interactive menu interface which allows you to choose a MultiValue emulation (Reality, mvEnterprise, mvBase or D^3) for a new database. This automatically enables data, keyword and item-id case-insensitivity if appropriate, sets the frame size to an appropriate value, and changes the default user operating environment to be compatible with the selected emulation (To find out which predefined environments are available and in which a particular option is set, use **SSM option 4** (Define Environment Settings) or the **DEFINE-ENVIRONMENT** TCL command.).

The **VendorEmulation** database configuration parameter specifies the MultiValue emulation. If set in the master database configuration file, it sets the default value for this **mkdbase** menu option. The **SYSTEM(102)** function (DataBasic and Proc) returns the currently selected emulation.

Case-insensitivity

Data Case-insensitivity

This can only be used if enabled both for the database with the **DateCaseControl** database configuration parameter and for a particular user with the **DATA.CC** environment option. Once enabled, it is possible to select or deselect it with either the **CASE** TCL command or the DataBasic **CASING** statement. If data case-insensitivity is enabled at both levels, it is initially selected for that user.

Features made case-insensitive
If data case-insensitivity is enabled, the following features become case-insensitive:

- DataBasic relational expressions.
- The following DataBasic functions and statements. **Functions: CHANGE**, **COMPARE**, **CONVERT**, **COUNT**, **DCOUNT**, **FIELD**, **FMT**, **GROUP**, **ICONV**, **INDEX**, **LOCATE**, **OCONV**, **TRIM**. **Statements: CONVERT**, **FIND**, **IF**, **INPUT**, **INPUT@**, **UNTIL**, **WHILE**.
- **SORT** function. The *sortOrder* parameter now accepts additional options to control data case-insensitivity.
- The Proc **IF** and **IF (Multivalued)** commands use case insensitive string comparisons.
- In English selection clauses, case is ignored when comparing attribute values.

Notes:

- Because index definitions are based on English, this also applies to indexes.
- If attribute 8 contains any pre-processor codes, case is also ignored when comparing item-ids.

The **SYSTEM(101)** function (DataBasic and Proc) returns the current state of data case-insensitivity. In D3 mode (set with the **$OPTIONS** statement) the **SYSTEM(28)** function returns the current case-insensitivity setting for DataBasic.

TCL Commands
The **SEARCH**, **ESEARCH** and **SSEARCH** commands now accept additional options to control data case-insensitivity.

Line editor
The editor commands that search for strings (**DE**, **L**, **R**, **SP** and **TR**) can now operate in both case-sensitive and -insensitive modes. The required mode can be specified when starting the editor (the default is the current data case setting) and changed while editing with new **CI**, **CS** and = commands. See *Search Strings* for more details.

Screen Editor
The screen editor commands that search for strings (**L** and **S**) can now operate in both case-sensitive and -insensitive modes. The required mode can be specified when starting the editor (the default is the current data case setting) and changed while editing with the new = command.

ME Development Editor
The ME editor commands that search for strings (**/**, **L** and **R**) can now operate in both case-sensitive and -insensitive modes. The required mode can be specified when starting ME (the default is the current data case setting) and changed while editing with new **CI**, **CS** and = commands.

Keyword Case-insensitivity

This can only be used if enabled both for the database with the KeyCaseControl database configuration parameter and for a particular user with the **KEY.CC** environment option. If enabled, the following features become case-insensitive as regards keywords:

- The contents of account and file definition items (local, synonym and remote).

Note: Although the user can use mixed case when editing a D-pointer, attribute 1 is converted to upper case when saved.

- The contents of **command** and **command synonym** definition items.

Notes:

- This does not currently apply to Proc commands, which must be entered in upper case.
- The case-sensitivity of TCL commands called from Procs and TCL macros depends on the item-id case sensitivity of the account's master dictionary.

- String keyword parameters in the following DataBasic functions and statements: **COMPARE** function (*alignment* parameter), **LOCATE** statement (*sequence* parameter), **SORT** function (*sortOrder* parameter), **TRIM** function (*type* parameter), **UNTIL** statement (*limit-expr* parameter), **WHILE** statement (*limit-expr* parameter).
- In the **ICOUNT** and **ISELECT** TCL commands, the case of alphabetic comparison operators is ignored.
- English conversion codes (wherever used) and their operators and operand keywords.
- Proc commands and their operators and operand keywords.
- Command definition items (attribute 1).
- Data definition items (attributes 1 and 9).

The following features are always available and do not have to be enabled:

- A new TCL command, **COMPILE**, provides a case-insensitive alternative to the BASIC command. This is functionally identical to BASIC, except that the case of keywords, variables and labels is ignored; for example, the variables FRED, fred and FrEd are treated as a single variable.
- DataBasic debugger. The **/** command now allows you to enter variable names in any case.

Item-id Case-insensitivity

This is enabled by selecting the appropriate option from the mkdbase menu when creating a database. To set the default to case-insensitive, either set the **IIDCaseControl** database configuration parameter in the master configuration file or set the **REALCASEINSENSITIVE** host environment variable.

Note: You can change an existing database to be item-id case-insensitive when you rebuild it by using **mkdbase** with the **-r** and **-c** options.

- If item-id case-insensitivity is enabled on a database, you can create files, accounts and directory views with case-sensitive item-ids. The INSENS.CREATE.FILE environment option specifies the default. This may be overridden with two new options: U (case sensitive, as on Unix) and W (case insensitive, as on Windows).

- Files, directory views and accounts are marked as case-insensitive by the addition of a D-pointer option (I) as the second character in attribute 1. The **I** option can only be set when the file or account is created, or with the **CONVERT-FILE** command; it cannot be changed by editing the D-pointer.
- Directory views. On UNIX hosts a directory view is made case-insensitive by making all item-ids lower case in the Reality file. As a result, all host files created as Reality items will have lower case names and only host files with lower case names are visible.

Note: Since Windows has a case insensitive file system, directory views are unchanged on this type of host.

The following item-id case-insensitivity features are also available:

- **SYSTEM(100)** function (DataBasic and Proc). This returns whether the database supports case-insensitive item-ids.
- The **USER**, **ISTAT**, **HASH-TEST** and **LISTFILES** TCL commands display whether the file contains case-insensitive item-ids.
- A new TCL command, **CONVERT-FILE** allows you to convert a file or an account MD from item-id case-sensitive to case-insensitive and vice versa. It is available in all user accounts, but can convert an account MD only if run from the SYSMAN or SYSPROG account.
- Elements 104 to 107 of the DataBasic **SYSTEM** function return the state of item-id case-sensitivity for the SYSTEM, NETWORK, USERS and SECURITY system files.
- The SYSTEM(108) function returns whether the host operating system is case-sensitive or -insensitive.

For information about how item-id case-insensitivity affects Reality, refer to *Case Sensitivity*.

Indexes

- When an index is created, the current settings for data and item-id case-sensitivity are saved with the index to ensure correct operation.
- Changing the item-id case-sensitivity of a file with the **CONVERT-FILE** command will invalidate any indexes. The indexes must be deleted and recreated to restore correct operation. The **VERIFY-INDEX** TCL command reports if the item-id case-sensitivity for an index does not match that of its file.

SQL for Reality

Although Reality files can now be case-insensitive as regards item-ids, SQL for Reality currently remains case-sensitive. The **SQLM** command has been enhanced to ignore case-insensitive files (and display an error message to that effect) when creating or regenerating SQL tables. A new option (W) allows you to force SQLM to convert case-insensitive files, but its use is not recommended because SQL access to Reality files is always case-sensitive.

DataBasic Environment Control

This allows the run-time behaviour of a DataBasic program or subroutine to be determined by a combination of two factors:

- The run-time version. There are two ways of setting this:
  - By using the compiler associated with the required version of Reality - the compiler version is embedded in the compiled code.
  - By specifying run-time version at the beginning of the source code with the **$OPTIONS** statement; the code produced will then behave as if it is being run on an earlier version of Reality. Note that can only be done if you use the V15 compiler or later; also, code compiled in this way can only be run on V15 or later.
- The MultiValue compatibility mode (if any) set at the beginning of the source code with the **$OPTIONS** statement.

Each combination of compiler version and MultiValue compatibility mode sets a unique combination of compatibility switches that control individual DataBasic features. These allow Reality V15 to work in a similar way to both earlier versions and other MultiValue systems, so that existing applications can run with the minimum of modification.

Logon Programs

- These could previously only be Procs, but it is now possible for any TCL command to be used; for example, a cataloged DataBasic program, synonym command, macro, sentence or menu. This applies to all types of Logon program: user, account and TCL inhibit.

  This feature is enabled on a user-by-user basis by setting the LOGON.PROGS environment option.
- User and Account logon programs. In previous versions of Reality, if both the user and the specified or default account had a logon program, only the user program was executed. If this feature is enabled and both the user and the account have logon program, when the user logon program completes, the account logon program is run.

  This feature is enabled on a user-by-user basis by setting the ACCOUNT.LOGON environment option.
- Global Logon Program. This allows the definition of a global logon program that runs when any user logs on. The program must be defined as an item called GLOBAL-LOGON in the MD of the GLOBAL.MD account.

TCL Macros

TCL macros provide an alternative to Proc for simple batch tasks. Four types of macro are available:

N-type macro — These allow the execution of a sequence of TCL commands. Any input required from the user must be appended as values to the command concerned. When run, TCL options supplied on the command line are applied to the first command in the macro.

M-type macro — This is similar to an N-type macro, but presents each command to the user for editing before execution.

S-type sentence — This is similar to an N-type macro, but supports different item formats for compatibility with other MultiValue systems.

TCL menu — This type of macro presents a list of options to the user and runs the TCL command associated with the selected option.

The following macro-only commands are provided:

DISPLAY statement

This displays a message to the user. The message can include literal text, the output of one or more TCL commands and various control sequences. An option allows the user to specify which MultiValue video codes are used (to simplify migration).

PROMPT statement

This displays a message to the user in the same way as the DISPLAY statement, but then waits for the user to enter C (continue) or Q (quit). If Q is entered, none of the subsequent commands are executed.

COMMENT statement

This is similar to the DISPLAY statement, but uses a different syntax.

REM statement — This allows remarks to be inserted into macros. All text which follows, up to the end of the line, is ignored.

Macros are items in the MD of an account. The user can create them with any of the Reality editors, or by using the **.C** TCL Stacker command to save commands from the stack as an S-type sentence. If a different type of macro is required, the user must use one of the editors to modify the saved item.

Macros of any type can be run from the TCL command line. In addition, an S-type macro can be loaded onto the TCL stack using the **.X TCL Stacker** command and the commands it contains executed from there. This is intended primarily for executing macros that were saved from the stack; any secondary input is ignored.

Other MultiValue Features

MultiValue Environments

- Support for the 'DATA' prefix to a filename.
- Predefined V15 environments have been provided for Reality, D3, mvEnterprise and mvBase. UniVerse, UniVision and UniData environments are now available. For details of which options are set in these, use SSM option 4 (Define Environment Settings) or the **DEFINE-ENVIRONMENT** TCL command.
- The TCL stacker command now has UniVerse, UniVision and UniData modes, set in the operating environment.
- The DataBasic **$OPTIONS** statement now accepts keywords representing UniVerse, UniVision and UniData systems. For each of these, an appropriate symbol is defined to control condition compilation with the **$IFDEF** and **$IFNDEF** statements.
- The **mkdbase** host command now allows you to choose UniVerse, UniVision and UniData emulations.

- A new **DEFAULT.TO.ALL** environment option allows the **COPY**, **CP**, **CT**, **ECOPY** and **QSELECT** TCL commands to behave in the same way as on D3 (if the item-list parameter is omitted, the default is *: all items). This option is set in the V15 D3 predefined environment.
- Predefined operating environments have been added for Reality V15, PICK, R83 and AP.

TCL

The information displayed by the **WHO** command and user exit **U50BB** depends on the selected MultiValue environment; for details, see *MultiValue Compatibility*.

English

- English now provides default attribute definitions that allow you to specify any attribute in an English command; for details, see the topics *Data Definition Item* and *MultiValue Compatibility* in the *English Reference* section of the documentation.
- Certain characters have special meanings when used in an English HEADING or FOOTING modifier. A new **NOOLDHEADER** environment option is set in the V15 MultiValue predefined environments to inhibit this behaviour.

Proc

- The Proc **A** command now allows you to specify the number of characters to be extracted from a parameter in the input buffer.
- A new **PQN.ABS.S** environment option changes the behaviour of the PQN Proc **S** command so that it will insert empty parameters, if necessary, to allow the pointer to be set to the specified position. This option is set in the V15 D3 predefined environment.

DataBasic

- **TCLREAD** statement. This is functionally the same as the SENTENCE function, but returns its result in a variable supplied by the programmer.
- **CHANGE** and **SWAP** functions. These now accept two additional parameters that allow you to specify the number of occurrences to replace and the occurrence with which to start.
- **EREPLACE** function. This is identical in operation to the **CHANGE** function.
- **DISPLAY** statement. This is identical in operation to the **CRT** statement, but is only available in AP, D3, MVBASE, PICK and R83 modes (set with the **$OPTIONS** statement).
- File error handling. In some MultiValue compatibility modes the THEN or ELSE clause for certain file handling statements is optional.
- The **TRIM** function has a new type option (M). This is similar to the "R" type, in that it replaces repeated occurrences of the specified character with a single character, but it does not remove leading and trailing occurrences.

  In D3 compatibility mode (set with the **$OPTIONS** statement), the "R" type functions in the same way as "M".
- **SYSTEM(98)** function. This returns a number representing the current video emulation; that is, the MultiValue vendor used to map the DataBasic @(-*n*) function within the current program module (determined by the MultiValue mode set with $OPTIONS).

It is also possible to set the video emulation for the current program module by using the ASSIGN or SYSTEM(98) statement.
When called from Proc, SYSTEM(98) always returns 0 (Reality).

- When using the **INPUT@** statement, a format string can now start with the letter "M" (for compatibility with other MultiValue systems).
- SYSTEM function: In Reality and D3 modes, if an item read fails because the item is locked, **SYSTEM(0)** now returns the number of the port holding the lock.
- **DQUOTE** and **SQUOTE** functions: In D3, AP, PICK and R83 modes, these functions search a string for a substring enclosed in quotation marks.
- **EXECUTE** statement:
  - In D3 mode, the SETTING clause returns error numbers only, rather than error numbers and messages.
  - The executed command can now be a dynamic array that includes responses to input prompts generated by the command.
- When called from DataBasic with **ICONV** or **OCONV**, the information returned by user exit **U50BB** depends on the MultiValue mode set with the **$OPTIONS** statement. For details, see *User Exit Conversions*.
- In D3 mode, strings containing only arithmetic unary operators and decimal points are treated as numeric.
- In D3, PICK/R83 and AP modes, THEN/ELSE clauses are not required in the **OPEN**, **READ**, **READU**, **READV**, **READVU**, **MATREAD** and **MATREADU** statements. If the statement fails, and there is no THEN or ELSE clause, the result (returned in the appropriate parameter) is set to null.
- **DIV**, **SADD** and **SSUB** functions: Provide alternative ways of performing arithmetic operations.
- **MSLEEP** statement: This causes a program to sleep for a specified number of milliseconds.
- **OUT** statement: Outputs raw data to the display.
- **CHANGE** statement: This changes the contents of a variable by replacing all occurrences of a substring with a new substring. (See also **CHANGE** function.)
- **TRANS** function: Extracts data from a file.
- **LEFT** and **RIGHT** functions: These extract substrings from the beginning and end respectively of a string.
- **FOLD** function: This has been enhanced to provide greater compatibility with other MultiValue systems.
  - If omitted, the fold width now defaults to 25.
  - Fold widths can be specified in a dynamic array.
  - The delimiter can now be specified.

- External functions: In additional to internal and external subroutines, Reality V15 allows you to write your own functions. These are similar to external subroutines, but return a value to the calling program and are called in the same way as intrinsic functions. The must be defined in a separately compiled and cataloged program module and declared before use in the calling program. New **FUNCTION** and **DEFFUN** statements are now available and the **RETURN** statement has been enhanced to return a value.
- Logical expressions. In D3, PICK/R83 and AP compatibility modes (set with the **$OPTIONS** statement), a complex logical expression are "short circuited" if a sub-expression evaluates to false. Subsequent sub-expressions are not evaluated. For details, see the **TRUE.BOOL** compatibility switch matrix setting.

Databasic and Proc

- **SYSTEM(94)** function. This returns a unique identifier based on the date and time, with a suffix to make this unique if necessary. In D3 compatibility mode this feature is available as **SYSTEM(19)**.
- **@ function**. Additional extended cursor addressing codes are now available in D3 and MVBASE compatibility modes (set with the **$OPTIONS** statement).
- **SYSTEM function**. D3 and MVBASE compatibility modes (set with the **$OPTIONS** statement) are now available.

Other Enhancements and Changes

Supported Platforms

The latest version of Reality adds support for:

- Windows 7 and Windows Server 2012 and 2008 (32-bit and 64-bit)
- CentOS 6 and 5 (Linux) (64-bit only)
- Solaris 11 (for SPARC 64-bit)

Removal of Windows-specific SQL Maintenance

The WinSQLM client component has been discontinued.

Partial D-pointers

If a File or Account definition item has less than 10 attributes, default values are used for the alignment and item-id column width (attributes 9 and 10).

mkdbase

- The **mkdbase** host command now accepts a switch (-h) that allows the hash type to be set.
- The **mkdbase** host command has an interactive menu interface which allows you to choose a MultiValue emulation (Reality, mvEnterprise, mvBase or D^3) for a new database. For more information, see Multi-Value Compatible Databases above.

TIPH Licences

A TIPH licence similar to the current Despooler licence is now available.

TCL

- The stacker mode where reused commands are moved to the top of the stack has been extended to include commands that are re-entered at TCL. See **.** (Dot - TCL Stacker Recall Command) and **SET-STACK**.
- TCL macros now combine options on the command line with those specified in the first line of the macro.
- The **ACCOUNT-RESTORE**, **M-A-R** and **SEL-RESTORE** TCL commands now accept a **W** option that specifies that the save is from a case-insensitive database.

Global TCL Commands

When a TCL command is run, the verb was previously searched for in the MD of the current account and then, if not found, in the file specified as the user's "Alternate Verbs File" (set in the user's security profile). This feature adds a third location - the MD of the GLOBAL.MD account.

An additional feature of this allows the user to bypass the local MD and the alternate file, and force selection of the global command. This is done by prefixing the command name with a tilde (~). By using this, the user could write a macro with the same name as a global command, but call the global version from with that macro.

A new environment option (INHIBIT~) allows the user to disable the latter feature, so that local commands with names beginning with tildes can still be run.

DataBasic

COMPARE Statement

This new statement allows you to compare two dynamic arrays.

PERFORM Statement

The performed command can now be a dynamic array that includes responses to input prompts generated by the command.

Conditional Compilation

$IFDEF and **$IFNDEF** constructs allow the programmer to mark sections of code for inclusion or exclusion depending on whether a symbol has been defined (using the **EQUATE** statement or a new **$DEFINE** compiler directive). The **$UNDEFINE** statement allows you to undefine a symbol that was previously defined with **$DEFINE**.

Note: When you select a MultiValue compatibility mode with the **$OPTIONS** statement, an appropriate conditional compilation symbol is automatically defined.

The **$TRUE**, **$T**, **$FALSE** and **$F**conditional compilation symbols provided by mvEnterprise are also supported.

EQUATE Statement

The **EQUATE** statement will now accept the LITERALLY keyword, used with a value enclosed in quotes. This allows the substituted value to contain spaces and multiple statements (separated by semicolons).

ACCESS Function

A new element (34) has been added to the **ACCESS** function. When this is used in a dictionary subroutine, **ACCESS** returns a string containing the next conversion code, if any. This makes it possible for DataBasic to emulate certain historic types of user exit.

Dynamic Array Functions

The following new functions are introduced:

- **ABSS** function

  Generates the absolute (positive) numeric values of the elements in a dynamic array.

- **ADDS** function

  Adds the elements in two dynamic arrays.

- **DIVS** function

  Divides the elements in one dynamic array by those in another.

- **MODS** function

  Divides the elements in one dynamic array by those in another, calculating the remainders.

- **MULS** function

  Multiplies the elements in two dynamic arrays.

- **NEGS** function

  Negates the elements in a dynamic array.

File Information Functions

Two new DataBasic functions return information about a specified Reality file:

- The **FILEPATH** function returns the path of the file referenced by a specified file variable.
- The **FILEINFO** function returns other information about the file referenced by a specified file variable.

Named Common Sections

Two new TCL commands are available:

NC.LIST This displays a list of the named common areas that have been defined and their current states.

NC.RESET This resets one or more named common areas.

In addition, when you log to a different account, any open files referenced by variables in named common sections are closed.

Debugger

The debugger displays a runtime error message if a variable is referenced before it has been assigned a value. The form of the message now includes the variable name, and depends on whether the variable is a simple variable, a vector entry or an array entry.

Default Compiler

The default DataBasic compiler is a new version that supports all the new features described in this topic. Note that code generated by this compiler cannot be run on earlier versions of Reality.

History Enhancements

The DataBasic history feature provides a mechanism to log history data to allow reporting and analysis of the usage of DataBasic programs, subroutines and external user functions. You create a separate HISTORY data section in your source files, and this section is used by the system to log the history usage of the code whenever it is complied and catalogued.

When any local source item is compiled and catalogued, a history log item with the same DataBasic program name is automatically written to the HISTORY data section of the source file (if such a section exists). Any previous history item is overwritten. The log item comprises the attributes shown in the following table (compilation is logged in attributes 1-7, cataloguing is logged in attributes 8-14).

Attribute	Description
1	Date of successful compilation (internal format)
2	Time of successful compilation (internal format)
3	PLID of successful compilation
4	ID of user requesting compilation
5	Account of successful compilation
6	Location of source (filename - including the data section, if appropriate)
7	Source item ID
8	Date of successful catalog (internal format)
9	Time of successful catalog (internal format)
10	PLID of successful catalog
11	ID of user requesting catalog
12	Account of successful catalog
13	Location of executable object (filename)
14	Executable item ID

The following dictionary definition items are supplied in the dictionary of the global POINTER-FILE. If cataloguing to the global POINTER-FILE then an item is created in the POINTER-FILE,HISTORY data section in addition to any in the source's HISTORY data section.

- DH.COMP.DATE - Date of successful compilation
- DH.COMP.TIME - Time of successful compilation
- DH.COMP.PLID - PLID of successful compilation

- DH.COMP.USER - ID of user requesting compilation
- DH.COMP.ACCT - Account of successful compilation
- DH.CAT.DATE - Date of successful catalog
- DH.CAT.TIME - Time of successful catalog
- DH.CAT.PLID - PLID of successful catalog
- DH.CAT.USER - ID of user requesting catalog
- DH.CAT.ACCT - Account of successful catalog

Note: By default, nothing is logged; the user has to enable compile/catalogue history logging (by adding the HISTORY data section).

Proc Error Messages

If a Proc is called from another Proc with the **P** command, the result might not be as expected; in particular, on completion of the called Proc, it will not return to the calling Proc. In this version, this problem can be harder to identify and solve, in that a Proc could call a TCL Macro which calls a Proc, etc., so error messages have been added to warn the user of this situation. On emulations, these are enabled by default; they can be enabled or disabled as required with the TRAP.EMBEDDED.PROC environment option.

Global POINTER-FILE

A new account will no longer contain a Q-pointer to the global POINTER-FILE, so it is no longer necessary to delete this before creating a local pointer file. Note however, that to access the global pointer file, you must include the account name (SYSFILES). For example:

```
LIST /SYSFILES/POINTER-FILE
```

Administration

Reality's command logging features (Audit and Support logs) have been enhanced to give the administrator greater control over what is logged and how the information is presented. For details, refer to the descriptions of the **AutditTclLog**, **SupportTclLog**, **SizeTclLog** and **TimeTclLog** database configuration parameters.

System Information

Two new commands, **SYSINFO** and **WHAT**, provide information about your Reality system that can be used for diagnostic purposes.

Installing Updates

- The way in which Reality updates are installed on Windows hosts now works in a similar way to on UNIX. This permits the installation of a single update if necessary without having to install a complete service pack. For details, refer to the *Installing Updates* section in the *Reality on Windows Installation Guide* and the **install_fix** host command.
- It is now possible to remove Windows updates with **install_fix**. **Note:** After removing an update, you must run SYS-UPDATE on each database to update the system files.

- On UNIX, it is no longer necessary to provide **install_fix** with the full path to the required update or the directory containing the updates.

Auto File Size

Dictionaries can now be marked for auto file size.

Documentation

Online Documentation

- The appearance of the Online Documentation has changed, although the content is substantially the same. In particular, the navigation features (Contents, Index, Search and Glossary) work slightly differently. For more information, see the *How to use the Online Documentation* topic.

Restrictions

This section lists the restrictions that were current when Reality V15.0 was released. For the latest information, refer to the Reality pages on the Northgate portal (www.northgate-is.com/reality).

All Versions

- File triggers can currently only be associated with file data sections.
- Shadow database cannot currently use partition databases constructed from standard host files on different file systems (see *Types of Database*).

On-line Documentation

If the documentation is installed on the local file system and you are using Internet Explorer, you will receive a number of security warnings regarding active content. These can be avoided by installing the documentation on a web server (recommended), or by selecting the **Allow active content to run in files on My Computer** option (you can find this in **Tools** | **Internet Options...**, on the **Advanced** tab under **Security**).

No Longer Supported

GUI Administration Tools

The GUI Administration Tools are no longer delivered as part of the Reality package, and existing installations will no longer be supported.

Fault Resolutions

Reality V15.0 includes resolutions of the following faults:

Fault Number	Description
084502	Add Am,n feature to Proc.
085003	The result of a Boolean expression may not be one or zero, but will be true or false.
085153	TCL man verb can report unassigned variable.
085170	Spin locks need to be able to support 32bit PLIDs.
085198	Problems with Install_Fix -a.
085200	SYSPROG-PL DBSTART.TEMPLATE syntax error.
085203	Migration of MV D-Pointer w/o justification in attribute 9 can cause English to display data with extra line feeds.
085221	When attempting to resync failsafe pair TLMENU gives message "Clean log clogxxx" cannot be found in event.log.
085245	The remote tape service is being stopped by windows service control manager due to invalid current state 0 error.
085249	Deleted account names are still visible in RealEdit.
085255	INPUT @(1,1):X,11:'*11' Gives error message 44.
085261	MD conversion does not handle an embedded decimal point.
085262	The T command in PQ proc causes screen positioning to be ignored.
085266	SLEEP "00:00" not sleeping.
085267	realdbc cannot cope with filenames that contain 085267 null characters.
085271	After upgrading to solaris 9 and loading gcc libs 3.4.6 npu and plidid abort.
085275	T-STATUS (V displays character 0xFF at the end of each line.
085281	Cannot install UNIX-Connect from the Reality V14.0 CD without a /usr/realman link or directory.
085282	Performing a program which has opened a file with a named common variable. If a second program opens the file, the lock will remain set even if released.
085291	Reality save does not benefit from compression if save is encrypted.
085297	Reality save aborts on restored MultiValue account.
085316	DCOUNT of null string value returns 1.
085328	Restore and verify report error if tape record is smaller size than that indicated in tape label.

Fault Number	Description
085330	CREATE-ACCOUNT takes option as account name.
085336	RECOVER-FD of deleted D-Pointer appears to work but does not recover file.
085348	Errors - rirchilddeath: wait for child failed.
085368	Restore of MV case-insensitive file causes Reality to crash later.
085372	QSELECT does not go down to subvalue level; stops at value level.
085373	SEARCH, ESEARCH and SSEARCH verbs are not case-insensitive.
085460	Cannot re-link v1.5 without answering 100's of prompts.
085482	Unix-Connect Session Manager loses incoming connections.
085488	Reality processes failing to logon with "Can't open /dev/null".
085516	F-Correlative used in index definition cause errors in daemon.log file.
085532	Cannot disable realbreak.
085555	WHERE command is not displaying a 'D' before despooler ports.
085556	Documentation on autofilesize config file parameter is incorrect.
085589	Timezone on AIX is not correctly recognised.
085612	DataBasic compiler syntax change 11 to 14.
090250	Indexes can be created with conversions that reference Denat messages, these are not available to update/verify.
090285	Clean logs saved with incorrect cpio format.
090339	TLMENU from telnet on Vista. Problem is with using net config workstation from Vista gives access is denied message.
090359	Install_Fix reports a vague and meaningless message when you install a fix for the wrong release.
090361	On define-environment screen, compare with column is left blank if default environment doesn't exist.
090390	RealEdit does not recognise new v14.0 commands like pause/wait.
090398	CREATE-FILE with no parameters creates dictionary with modulo 7 - should be 1. data section is correct (7).
090401	Running DEFINE-ENVIRONMENT form an account other than SYSMAN gives "is not on file" not "must be run from SYSMAN".
090402	CREATE-ACCOUNT allows account names containing slashes.
090408	Servers on Aix do not use Timezone and daylight saving corrections.
090413	Compilation errors found within a BEGIN CASE block report the correct source text but report the line number of the BEGIN CASE line.

Fault Number	Description
090417	ENGLISH-TUTORIAL has items missing from its files.
090426	COPY {filename} {item} (AT or CT {filename} {item (A either leaves the user in the editor or aborts.
090427	Turning debug tracing (such as rcstrace) causes various problems with Reality related functionality such as POVF and realdbck.
090433	IF REMOVE_TYPES = 1 THEN gives bad statement error.
090436	DataBasic checksum function stops on 1st SM delimiter.
090437	Random aborts if new compiled DataBasic object run on old release.
090438	Entering numbers into the TCL options field is inconsistent.
090440	Inappropriate message logged in daemon.log for all non-network connections (i.e.console).
090443	English is giving an invalid SOUNDEX result.
090446	Server connection fails to validate password.
090452	LOADENT as used by mkdbase cannot process extended reality T-DUMP format tapes.
090460	Unable to change system privilege level of remote sessions.
090464	Can't use IN_STR as variable, while CRT_STR is OK.
090475	Concatenate of SENTENCE( ) gives rwsblkxtnd: invalid workspace size requested.
090476	Sort of a specified attribute or value returns a null.
090481	For MV, allow include statement where comma is between file-specifier and item-name.
090501	Memory corruption if error message missing.
090503	SET-ENVIRONMENT should display current environment.
090509	TCL-inhibit program causes continuous loop.
090512	Case-sensitive files are restored from a MultiValue save as insensitive.
090520	The stacker .Q command doesn’t work.
090527	Writing to a DIRVIEW with an item id containing characters that are illegal in the host operating system is not handled correctly.
090531	CATALOGing DataBasic in MV mode can corrupt D-pointer.
090541	Keyboard echo can become disabled.
090553	SORT never completes.
090560	Windows netadmin utility does not provide a way to clear a field.
090563	Unix Connect telnet_lbs coredumps on Solaris 10.

Fault Number	Description
090564	cdinstall option 3 fails to install.
090577	M-A-R fails with signal (11) in frame 2102.
090578	DataBasic shows wrong line number in error for single END in IF statement.
090586	Files with names containing slashes are not accessible using direct Q/Pointers.
090672	Proc transfers can fail in case insensitive account.
090756	GET-INDEX does not close index control block.
090778	Unix Connect netadmin on Unix does not handle sock interface entries.
090779	Unix Connect socket i/f listeners can miss incoming connections.
090785	SqlExtendedFetch can 'hang' doing type sql_fetch_bookmark example vs2005 project available to reproduce.
090801	Setting rgstrace can cause core dump in rgsescreatefilerek.
090806	Reality can hang on windows when rgs tracing is enabled.
090824	Cannot call user function without assigning result.
090845	New risc client cannot find default account.
090847	rw_hspace displays "horizontal spacer" in Firefox.
090849	Linux home escape sequence crashes Reality on windows.
090855	netadmin - defaults doesn't allow selection of more than 20 databases.
090864	Menu PCSM displays error if number > 5 is entered.
090865	MSG ![ownport] no longer sends message to screen.
090870	BREF produces unsorted list.
090872	Capture fails if SYSDATA file does not exist.
090873	Core dump when doing BREAK 1000.
090874	Fieldread behaves oddly.
090877	TRUNC falls over rather than showing error.
090881	Select a non-existent index causes fallover.
090889	Sqlsrvr dies after 30000 operations.
090914	Unable to start TLMENU after abortive patch load.
090915	Trap at virtual code 2373,295.
090917	Corruption in middle of returned key list.
090956	mkdbase fails "getnextchar: illegal seq ff 00".
090957	Installing fixes on Database can be very slow.

Fault Number	Description
090960	Incorrect tape type for tape image can cause core dump.

Third Party Products

The following third party products are used within Reality:

- GNUmalloc (GNU Software Foundation);
- zlib compression library (GNU Software Foundation).
- DES Encryption library (Eric Young - eay@cryptsoft.com).

The following third party products may be included with Reality (depending on the operating system and the features selected):

- Perl scripting environment (GNU Software Foundation);
- Gzip compression software (GNU Software Foundation);
- GNU C-complier (GNU Software Foundation);
- Gdb Debugger (GNU Software Foundation);
- Adobe Acrobat document reader (Adobe Systems Inc.);
- TomCat web server (Apache Software Foundation).
- Jetty web server (Mort Bay Consulting).