



jBASE Product Release Notice

Product:	jBASE 5.2
Version:	5.2.31
Type:	Patch Release

Contents

Features 3

Components 5

Supported Platforms..... 6

Patch Details 7

Installation Instructions for jBASE 5.2.31 8

Incidents Addressed in jBASE 5.2.31 Release 10

Patches Incorporated in jBASE 5.2.31 Patch Release..... 11

Features



Spotlight on jBASE

Since its founding in 1989, jBASE has become known as a technological trailblazer in the MultiValue industry, offering unparalleled robustness, scalability and connectivity. 25+ years of hard work and determination has led to the evolution of a product line perfectly placed for today's and tomorrow's application needs. jBASE is positioned to migrate existing MultiValue applications (with functionality intact) to the new world of mainstream application development environments.

jBASE provides complete freedom in your IT environment by ensuring there are no barriers to your growth. The open architecture of jBASE is uniquely designed to allow a wide range of application development tools and backend databases to form part of a jBASE solution. If the answer to your business problem is a mainstream database or 3rd party application, then it is a simple task to integrate using jBASE.

Interoperability with Oracle, DB2, SQL Server

A fundamental design feature of jBASE is database independence. By means of an innovative, open and published interface (jBASE External Device Interface, jEDI), any jBASE developed application can read and write to whichever database is required for data storage just as simply as the jBASE database. This means a jBASE solution can be billed as an Oracle, DB2 or SQL Server solution without a major rewrite and at an economic cost. Major end users can match a corporate requirement for RDBMS compliance while still benefiting from thousands of hours of development in tailored applications written in BASIC. VARs can deliver applications to markets demanding RDBMS databases to capitalize on their investment and expand their business. Even the tough task of merging two companies with diverse IT strategies is easily accomplished by jBASE and its jEDI architecture. Couple this with the variety of operating platforms that jBASE runs on and it's plain to see that your options are endless.

Seamless Integration with Mainstream Development Tools

jBASE runs natively on an operating system so it does not incur the overhead or the inconvenience of running interpreted code in its own, enclosed, environment like most of the other Multi-Value databases. A jBASE program is compiled and catalogued as an operating system level executable that can be called from any other piece of executable code capable of calling external routines. Most importantly, once compiled, interoperability with third-party applications and drivers is seamless and native.

If you have BASIC code that is responsible for the routing of your fleet of trucks or for determining the best way to place goods in a warehouse or for assessing risk on a stock portfolio then you can now call it from any other environment – Java, VB.NET, C or C++ or even your favorite IDE.

Running natively with the host operating system ensures that new technologies are easily supported which allows many options relating to future direction. Should Java or .NET or any other tools be used to develop applications, jBASE programs will seamlessly fit alongside or as part of the same application. The design of jBASE means there is no environment or shield between jBASE applications and the underlying operating environment which makes for more elegant interoperability and optimum performance levels.

Future Proof your Application

The combination of powerful tools and a powerful database, together with truly open middleware means that, should you decide on jBASE for your application, you can rest assured in the knowledge that your investment is protected. Your data can be stored in jBASE or in any mainstream RDBMS or file system you require. Whatever the skill set of the developers in your organization, whichever technology you need to use, you can develop using jBASE safe in the knowledge it will be a future proof solution. There is no limit to what you can do with the technology provided by jBASE. The open architecture puts you in control and provides you with choice.

Release 5: Mainstream Benefits

The release of jBASE 5 marks a major new development for jBASE and brings Multi-Value users closer to the mainstream than ever before. This true 64-bit release delivers a new level of resilience through the introduction of the jBASE Dataguard suite. This collection of technologies uses the same concepts

as SQL Server, Oracle and DB2 to allow jBASE to operate non-stop as a technology platform.

Why 64-bit? 64-bit removes many of the limitations that have caused problems in the past, for example, Shared Library size on AIX, File handles on Solaris and file sizes on just about everything! As time marches on, the number of computer systems being built on a 32-bit architecture is dwindling and main manufacturers are only delivering 64-bit machines with 64-bit operating systems. While some MultiValue databases might have added 64-bit addressing to overcome 2GB file sizing, jBASE 5 is a complete 64-bit implementation of the database. There are no hidden 32-bit addressing limitations, allowing complex and large applications to be handled with ease.

Non-Stop Computing

jBASE 5 investment has been predominately in the area of Transaction Journaling which has been substantially engineered to provide an ever greater level of flexibility and robustness.

A number of new structures have been introduced to allow more precision in recovery. One of the major enhancements has been the addition of the concept of Checkpointing. This process periodically sets markers which record the fact that the database is at a known point at that specific time. The recovery modules can refer to these checkpoints to keep the system in a robust state.

Warmstart recovery is designed to enable a database to be returned to a stable, working position following a power failure in a similar fashion to mainstream RDBMS products. It automates the recovery of a system that has been improperly shutdown with no manual intervention required.

The Online Backup facility has been developed to enable system managers to perform necessary regular database backups while still allowing users the ability to perform updates on the database. This functionality also enables file restoration on a live system.

jBASE 5 delivers not only some of the best system integrity and data security features in the MultiValue database market today but ones that are on par with many mainstream RDBMS products. The new features are all about reducing downtime

which is arguably the goal of all data management solutions. For the first time, MultiValue solutions can be billed as non-stop in both application and database.

Java Development

New with jBASE 5 is a whole suite of functionality to benefit enterprise application development. Java Enterprise Edition (JEE) application development on jBASE 5 is supported by both the major JCA enhancements to jBASE Basic and new JDBC implementation.

This is the first time that the full JEE functionality has been supported on jBASE. And the implementation ensures the full Java Enterprise applications can be developed and deployed with ease. To round out the support for the developer the new jRemote Java functionality ensures that whatever the Java requirement, jBASE 5 can deliver. It enables the Java developer to make use of the rich functionality in existing BASIC code and exploit features such as jBASE 5 transaction processing and UTF-8 Internationalization.

For the technical at heart we have enabled Java to include JEE Connector Architecture (JCA) compliance, accept connections with JEE clients and manage the lookup, execution and transactional flow of Enterprise Java Beans (EJB), JMS queues and requests from Message Driven Beans (MDB). For those that prefer SQL, the jBASE query processor has been enhanced to accept SQL commands as well as jQL commands.

SQL Support

One of the main benefits of providing a SQL engine for jBASE is that the database can be used with external tools and APIs. SQL has many benefits that can be applied to the jBASE MultiValue, hierarchical database. In particular with jBASE, SQL allows users to query data where there might be tables within tables and no primary-key/foreign key relationship. This is an extreme advantage not available in most RDBMS systems. As well as integration with external API's, the jBASE SQL engine also allows SQL to be used to interact directly with jBASE files. For example, SQL can be used wherever jQL is used currently while the rich set of SQL functions allows the creation and manipulation of data tables e.g. inserting updating and deleting records.

Visit www.jBASE.com to discover an efficient and elegant way to move your information technology into the next generation.



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 databases including SQL, Oracle and Pick MultiValue systems.

ZUMASYS HEADQUARTERS

9245 Research Drive, Irvine, CA 92618 | US: 866-582-8447 | UK: 0808-189-3266 | sales@jbase.com

Copyright © 2015 jBASE. All Rights Reserved. DS-020315

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



www.jbase.com

Components

.Net OBJEX	Provides VB/.Net interoperability.
jRemote Net	Provides Native jBASE API for accessing jBC functions remotely

3rd Party Components

The jBASE 5.2 installation process includes installation of the following 3rd party components.

Java Runtime Engine

Built using the following Java versions:

HP-UX B.11.23 Itanium	Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0.04-_27_jul_2006_10_52)
AIX 5.3	Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0)
Solaris 10 SPARC	Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_12-b04)
Linux RH-AS5	Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_22-b03)
Windows 64-bit	Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_12-b04)

Internationalization Library

Required for Locale and Multi Byte character set support

- ICU 4.0.1

XML Library

Required for XML function support.

- XERCES 2.7.0
- XALAN 1.10.0

Supported Platforms

This specific Patch Release provides support for the following platforms:

AIX 5.3 (64-bit) *
HPUX Itanium 11.31 (64-bit)
Red Hat Enterprise Linux 5 (64-bit) *
Solaris 10 SPARC (64-bit) *
Windows (64-bit)

*This build is binary compatible with forward releases of the platform

(e.g. AIX 5.3. -> AIX 6.1 -> AIX 7.1)

*Running on Red Hat 6 required the SSL compatibility library to be installed; this can be done with the following command:

```
yum install openssl098e
```

Caveats

- jDP / Attunity Connect is not available for the 64-bit jBASE 5.2 releases

Compilation

jBASE 5.2.31 is brought to you by the following compilers;

Platform	Compiler version
AIX 5.3	XL C/C++ Version 8.0.0.18
Windows 64-bit	Visual Studio 2005 SP1
RH Linux AS5	gcc (GCC) 4.1.2 20080704 (Red Hat 4.1.2-44)
HP Itanium3	cc: HP C/aC++ B3910B A.06.22 [Nov 14 2008]
Solaris 10 SPARC	Sun Studio 11

Patch Details

All patches from previous jBASE releases have been included where appropriate in jBASE 5.2.31. All Patch details for the jBASE 5.2.31 release are provided in a separate Patch Summary document.

jBASE 5.2.1 patch release includes Patches PN5_20000 through PN5_20051

jBASE 5.2.2 patch release includes Patches PN5_20052 through PN5_20067

jBASE 5.2.3 patch release includes Patches PN5_20068 through PN5_20089

jBASE 5.2.4 patch release includes Patches PN5_20090 through PN5_20099

jBASE 5.2.5 patch release includes Patches PN5_20100 through PN5_20110

jBASE 5.2.6 patch release includes Patches PN5_20111 through PN5_20117

jBASE 5.2.7 patch release includes patches PN5_20118 through PN5_20123

jBASE 5.2.8 patch release includes patches PN5_20124 through PN5_20143

jBASE 5.2.9 patch release includes patches PN5_20144 through PN5_20153

jBASE 5.2.10 patch release includes patches PN5_20154 through PN5_20163

jBASE 5.2.11 patch release includes patches PN5_20164 through PN5_20184

jBASE 5.2.12 patch release includes patches PN5_20185 through PN5_20195

jBASE 5.2.13 patch release includes patches PN5_20196 through PN5_20202

jBASE 5.2.14 patch release includes patches PN5_20203 through PN5_20208

jBASE 5.2.15 patch release includes patches PN5_20209 through PN5_20221

jBASE 5.2.16 patch release includes patches PN5_20222 through PN5_20234

jBASE 5.2.17 patch release includes patches PN5_20235 through PN5_20247

jBASE 5.2.18 patch release includes patches PN5_20248 through PN5_20257

jBASE 5.2.19 patch release includes patches PN5_20258 through PN5_20271

jBASE 5.2.20 patch release includes patches PN5_20272 through PN5_20284

jBASE 5.2.21 patch release includes patches PN5_20285 through PN5_20294

jBASE 5.2.22 patch release includes patches PN5_20295 through PN5_20307

jBASE 5.2.23 patch release includes patches PN5_20308 through PN5_20326

jBASE 5.2.24 patch release includes patches PN5_20327 through PN5_20333

jBASE 5.2.25 patch release includes patches PN5_20334 through PN5_20345

jBASE 5.2.26 patch release includes patches PN5_20346 through PN5_20356

jBASE 5.2.27 patch release includes patches PN5_20357 through PN5_20363

jBASE 5.2.28 patch release includes patches PN5_20364 through PN5_20375

jBASE 5.2.29 patch release includes patches PN5_20376 through PN5_20387

jBASE 5.2.30 patch release includes patches PN5_20388 through PN5_20404

jBASE 5.2.31 patch release includes patches PN5_20405 through PN5_20424

Installation Instructions for jBASE 5.2.31

UNIX

Set the 'umask' to enable the correct UNIX permissions for the files about to be installed.

```
umask 0
```

Create the directory into which you are installing jBASE 5.2, eg :

```
mkdir /home/jbc52
```

This directory path will subsequently be used as the '\$JBCRELEASEDIR' environment variable setting.

[ensure the volume in which you are installing jBASE 5.2 has sufficient free disk space, approximately 500MB is required, plus additional space for any temporary files]

```
df -k [ will show the current disk usage within each UNIX volume ]
```

'cd' to the directory just created, eg:

```
cd /home/jbc52
```

Uncompress the appropriate 'tar.gz' file, eg for 64-bit AIX 5.3:

```
gzip -d AIX_53_JBASE_5231.tar.gz
```

Install the jBASE release using:

```
tar -xvf AIX_53_JBASE_5231.tar
```


Windows Systems

Run the installer, e.g. 'WIN_JBASE_5231.exe' and follow the on screen instructions/prompts shown.

Note: The Installer has been modified slightly so that if it finds a "config" directory in the install directory it will assume that its doing an "Upgrade" and will stop the telnet and jDLS daemons and then rename the existing config directory to "config_pre{jBASE Install Version}". e.g. "config_pre5.2.31"

It will then pop up a message box displaying this information.

Incidents Addressed in jBASE 5.2.31 Release

Ticket Number	Description	Patch Number
581490	Nested multi-value math functions would only operate on the first multi-value	5_20405
577321	SSELECT does not work with SAMPLE	5_20406
577371	SELECT DICT and ANY	5_20407
557436	SQL incorrect results when using date math with mv dates	5_20408
576952	jbase_agent -a user enhanced to allow environment changes in JAGENT_USER file	5_20409
-	Update website details, add new images and icon.	5_20410
584665	Repeated TRANS() function calls retains a handle on the file	5_20411
580200	jKeyCheck does not parse the list of products from the license key	5_20412
596448	SQLDESCRIBE fatal abort	5_20413
593006	Test Group Extract in jQL F-Correlative	5_20414
582544	PH-DELETE and PH-ALLOCATE required root/administrator privileges	5_20415
594005	Prevent 'Segmentation Violation' compiling embedded SQL	5_20416
600416	Z element of a format mask	5_20417
586371	long keys causing buffer overrun when jBASE displays a message.	5_20418
584047	[TJ] Prevent items being written to a logset AFTER the 'End of File' record	5_20419
585062	SYSTEM(20) is empty	5_20420
590579	SETPTR enquiry	5_20421
602933	Provide 'JBASE_DATE_FORMAT' environment variable	5_20422
601468	Q2Q pointers when JEDIFILENAME_MD points to somewhere outside the account	5_20423
600421	Prevent incorrect 'Free License Count' as shown by 'jlicensinginfo'	5_20424

Patches Incorporated in jBASE 5.2.31 Patch Release

Patch Number: 5_20405

Ticket Number: 581490

Description: Nested multi-value math functions would only operate on the first multi-value

When the MULS() and ADDS() functions are nested then they only used the first multi-value. This issue was only seen if REUSE() was used in the nested math function expression.

Previous Release Behavior:

```
0001 multivalues = "0.025]0.035]0.045"
```

```
0002 CRT MULS(ADDS(multivalues, REUSE(0)), REUSE (100000))
```

Running this program would return:

```
2500]0]0
```

Current Release Behavior:

The program now correctly returns:

```
2500]3500]4500
```

Patch Number: 5_20406

Ticket Number: 577321

Description: SSELECT does not work with SAMPLE

Previous Release Behavior:

```
jsh →SSELECT JDP SAMPLE 2
```

```
5 Records selected
```

Note that the SAMPLE works for SELECT, LIST, SORT and even SREFORMAT.

Current Release Behavior:

```
jsh →SSELECT JDP SAMPLE 2
```

```
2 Records selected
```

Patch Number: 5_20407

Ticket Number: 577371

Description: SELECT DICT and ANY

Previously jQL did not support the throw away verb ANY.

Previous Release Behavior:

```
jsh -->LIST DICT JDP WITH ANY *A9 = "L"  
Error in Statement "LIST DICT JDP WITH ANY *A9 = "L"  
Parse Error: ANY.
```

Current Release Behavior:

```
jsh -->LIST DICT JDP WITH ANY *A9 = "L"  
PAGE 1 08:18:52 12 MAR 2015
```

DICT JDP.....

FIRSTNAME
LASTNAME
ADDR1
ADDR2
CITY
STATE
ZIP
HOMETEL
WORKTEL
EMAIL
HARDWARE
OS
NUMUSERS
SYSTEMTYPE

14 Records Listed

Patch Number: 5_20408

Ticket Number: 557436

Description: SQL incorrect results when using date math with mv dates

Previously it was not possible to limit a results set using an associated date, the returned results would fail if there were multi values present. This behaviour could be seen when trying to limit associated data using WHERE .

Previous Release Behavior:

PROMISE	Dictionaries	Promise_Entered_Date	Promise_Date
9387	001	A	A
001 16465	002	1	3
002 200	004	C;2;3	D;1
003 16469	007	D2	D2
	009	L	L
9383	010	10	10
001 16465]16900			
002 200]300			
003 16466]16454			

9386
001 16465
002 200
003 16466

9395
001 16465]17250
002 200]125
003 16466]16554

```
jsh danell ~\jBASE -->SQLSELECT RECID,Promise_Entered_Date,Promise_Date FROM PROMISE
```

```
RECID      Promise_En Promise_Da
-----
          9387 2013-01-28 2013-02-01
          9383 2013-01-28 2013-01-29
          9383 2014-04-08 2013-01-17
          9386 2013-01-28 2013-01-29
          9395 2013-01-28 2013-01-29
          9395 2015-03-24 2013-04-27
```

Selected 6 rows.

```
jsh danell ~\jBASE -->SQLSELECT RECID,Promise_Entered_Date,Promise_Date FROM PROMISE WHERE
(Promise_Entered_Date+1)=Promise_Date
```

```
RECID      Promise_En Promise_Da
-----
          9386 2013-01-28 2013-01-29
```

Selected 1 rows.

Current Release Behavior:

```
jsh danell ~\jBASE -->SQLSELECT RECID,Promise_Entered_Date,Promise_Date FROM PROMISE WHERE
(Promise_Entered_Date+1)=Promise_Date
```

```
RECID      Promise_En Promise_Da
-----
          9383 2013-01-28 2013-01-29
          9386 2013-01-28 2013-01-29
          9395 2013-01-28 2013-01-29
```

Selected 3 rows.

Patch Number: 5_20409

Ticket Number: 576952

Description: jbase_agent -a user enhanced to allow environment changes in JAGENT_USER file

JAGENT_USER file only contained User as key and password information. Added fields to JAGENT_USER file similar to those in SYSTEM file to set environment for that user.

Previous Release Behavior:

JAGENT_USER file only contained User as key and password information

Current Release Behavior:

Added fields to JAGENT_USER file similar to those in SYSTEM file to set environment for that user

```
0020 ESYSTEM_START
0021 prime                ← JBCEMULATE
0022 c:\jBASE5\data\MAIN  ← HOME
...
0026 c:\jBASE5\trigidx\lib ← JBCOBJECTLIST
0027 c:\jBASE5\global;c:\jbase5\data\MAIN ← JEDIFILEPATH
0028 c:\jBASE5\data\MAIN\MDJD ← JEDIFILENAME_MD
...
0037 ESYSTEM_END
```

Patch Number: 5_20410

Ticket Number: -

Description: Update website details, add new images and icon

Patch Number: 5_20411

Ticket Number: 584665

Description: Repeated TRANS() function calls retains a handle on the file

Original reported issues is, a file is created and written to then deleted and re-created and new data is written. However the program returns incorrect [deleted] results. This issue is caused by the files used in jBC TRANS being cached, previously the following behaviour would occur.

Previous Release Behavior:

Windows:

Process is unable to delete the first files as the transaction cache also has the file open.

Unix:

Process deletes file, create new but then returns the first record read from cached file pointer.

Given the following program...

```
0001 filename = 'testfile'
0002 EXECUTE 'CREATE-FILE ':filename:' 1 1'
0003 OPEN filename TO filevar ELSE STOP
0004
0005 rec = 'record 1'
0006 WRITE rec ON filevar, 'xyz'
0007 trans_rec1 = TRANS(filename, 'xyz', -1, 'X')
0008
0009 EXECUTE 'DELETE-FILE ':filename
0010
0011 EXECUTE 'CREATE-FILE ':filename:' 1 1'
0012 OPEN filename TO filevar ELSE STOP
0013
0014 rec = 'record 2'
0015 WRITE rec ON filevar, 'xyz'
0016 trans_rec2= TRANS(filename, 'xyz', -1, 'X')
```

...the variables 'trans_rec1' and 'trans_rec2' would contain the same data, i.e. 'record 1'.

Current Release Behavior:

New behaviour is to clear out any reference to the file from transaction cache when you call DELETE-FILE.

The variable 'trans_rec2' contains the correct value, i.e. 'record 2'.

Patch Number: 5_20412

Ticket Number: 580200

Description: jKeyCheck does not parse the list of products from the license key & to display correct Websession information

Previous Release Behavior:

The jKeyCheck command would display a list of all products instead of the actual products incorporated into the license key, and when used with a Websession key did not parse the key into number of Users and Systems.

Current Release Behavior:

The jKeyCheck command now displays the correct list of products, and parses the Websession Users and Systems correctly, e.g.

Key type	[standard]
Machine name	[webserver]
License type	[Web-session license]
Users	[3]
System(s)	[1]

Patch Number: 5_20413

Ticket Number: 596448

Description: SQLDESCRIBE fatal abort

Previous Release Behavior:

When there was a secondary index on the file, the command 'SQLDESCRIBE filename' aborted on Windows with a 'SQLDESCRIBE.exe has stopped working' dialog box and on Unix with a Segmentation Violation.

Current Release Behavior:

'SQLDESCRIBE filename' does not abort when the file has a secondary index.

Patch Number: 5_20414

Ticket Number: 593006

Description: Test Group Extract in jQL F-Correlative

Fix issue with G (group) conversion when used in F correlative, [CW59306] Previously the following would fail when the conversion in attribute 8 was called. 0001 S 0002 0 0003 TEST_DICT 0004 0005 0006 0007 0008 F;(G*1) 0009 L 0010 15 the following error message would be displayed. Error in attribute definition item TEST_DICT Error in F correlative: F Correlative stack underflow operator: Conversion Internal the issue was cause because this is being called via an F correlative. F;(G*1) jQL was expecting to see an attribute number before the group function. F;2(G*1) jQL has now been changed to always use the attribute number defined in the current dictionary item if nothing is supplied in the conversion. In the above example this will be the equivalent of doing... F;0(G*1)

Patch Number: 5_20415

Ticket Number: 582544

Description: PH-DELETE and PH-ALLOCATE required root/administrator privileges

Previous Release Behavior:

The PH-DELETE and PH-ALLOCATE commands could only be run by a user with root (or Administrator) privileges.

Current Release Behavior:

Any user can now run PH-DELETE and PH-ALLOCATE.

Patch Number: 5_20416

Ticket Number: 594005

Description: Prevent 'Segmentation Violation' compiling embedded SQL

Previous Release Behavior:

Prior to this patch './prnttst' would give a 'Segmentation Violation'

Current Release Behavior:

Basic program (prnttst.b)

```
DEFCE INT prnt_tst(VAR)
```

```
s = STR('abc',12768)
```

```
CRT BYTELEN(s)
```

```
x = prnt_tst(s)
```

C program (prnt_tst.c):

```
#include <jsystem.h>
```

```
int prnt_tst(VAR *str)
```

```
{char DeclareFile[100];
```

```
FILE *sqlfd;
```

```
struct jBASEDataAreas *dp;
```

```
sprintf(DeclareFile, "%s", "print_contents");
```

```
if ((sqlfd = fopen(DeclareFile, "w+")) == NULL)
```

```
{fprintf(stderr, "Cannot create '%s' file\n\r", DeclareFile);
```

```
return 3;}
```

```
JBASEfprintf(sqlfd, VAR_STRING_ADDR(str));
```

```
fclose(sqlfd);
```

```
return 0;}
```

Compile with :

```
jcompile -I $JBCRELEASEDIR/include prnttst.b prnt_tst.c
```

jsh support ~/PGM --> ./prnttst

38304

jsh support ~/PGM -->

Patch Number: 5_20417

Ticket Number: 600416

Description: The "Z" format mask element did not return the correct format for zero valued floating point numbers

The general form of the mask is: Mjn{m}{Z}{,}{\$}{xi}{c} j specifies the justification of the output data and will be one of: L indicating that the data is to be left-justified R indicating that the data is to be right-justified for this issue we are only concerned with {Z}, Z indicates that any leading zeros in the data are to be suppressed.

Previous Release Behavior:

Previously any zero valued floating point numbers would not return the correctly formatted string.

Test:

```
CRT DQUOTE(FMT(0.00, 'R2Z#10'))
```

```
CRT DQUOTE(FMT(0, 'R2Z#10'))
```

Output:

```
""  
"    "
```

Current Release Behavior:

Output:

```
"    "  
"    "
```

0.00 and 0 both have a value of zero so are not displayed but are correctly formatted.

Patch Number: 5_20418

Ticket Number: 586371

Description: long keys causing buffer overrun when jBASE displays a message.

Previous Release Behavior: Internally the issue was that jBASE had a 16k buffer to format any messages and was passed a 16k+ key. e.g. id = STR("x", 20000) READ rec FROM file,id ELSE STOP 203, id

Current Release Behavior: The current behavior is it does not core dump when passed a huge record ID.

Patch Number: 5_20419

Ticket Number: 584047

Description: [Transaction Journalling] Prevent items being written to a logset AFTER the 'End of File' record

Prevent logset corruption when switching logsets

To test : jlogadmin -R -l next whilst another process is updating items with Transaction Journalling enabled

Previous Release Behavior:

The writing process (Pid 0) is still running but has not written all data to Transaction Journal' may have been seen during a subsequent 'jlogdup' or 'LIST TJlogfile' involving the 'logset' switched to above

Current Release Behavior: This error no longer occurs and all issues encountered during testing have been addressed.

Patch Number: 5_20420

Ticket Number: 585062

Description: The SYSTEM(20) function returns '0' (zero)

Previous Release Behavior:

When the JBASEUNIQUE environment variable was set, the SYSTEM(20) function would return '0' (zero) for the last spooler job number.

Current Release Behavior:

SYSTEM(20) now correctly returns the last spooler job number regardless of whether or not JBASEUNIQUE is set.

Patch Number: 5_20421

Ticket Number: 590579

Description: SETPTR query does not show destination printer:

Previous Release Behavior:

Prior to this patch 'SETPTR ?' displayed 'NONE' for 'Destination printer'

To test : jsh -> SP-ASSIGN =STANDARD jsh -> SETPTR ? Unit Number : 0 Page Width : 132 Page Depth : 66 Top Margin : Bottom Margin : Print mode : - Spooled Output Default spool banner : "P#0000" Destination printer : /dev/lp0 Initial Job State : PRINT

```
jsh jbasetest ~ -->SETPTR ,,,,,,AT STANDARD
```

```
Unit Number :0
```

```
Page Width :132
```

```
Page Depth :66
```

```
Top Margin :3
```

```
Bottom Margin :3
```

```
Print mode : 1 - Spooled Output
```

```
Default spool banner : "P#0000"
```

Destination printer : "/dev/lp0"

Initial Job State : "PRINT"

jsh jbasetest ~ -->SETPTR ,,,,,AT STANDARD

Unit Number :0

Page Width :132

Page Depth :66

Top Margin :3

Bottom Margin :3

Print mode : 1 - Spooled Output

Default spool banner : "P#0000"

Destination printer : "NONE"

Initial Job State : "PRINT"

Current Release Behavior:

jsh jbasetest ~ -->SETPTR ,,,,,AT STANDARD

Unit Number :0

Page Width :132

Page Depth :66

jsh jbasetest ~ -->SETPTR ?

Unit Number :0

Page Width :132

Page Depth :66

Top Margin :3

Bottom Margin :3

Print mode : 1 - Spooled Output

Default spool banner : "P#0000"

Destination printer : /dev/lp0

Initial Job State : PRINT

Patch Number: 5_20422

Ticket Number: 602933

Description: Provide JBASE_DATE_FORMAT environment variable

Previous Release Behavior: None

Current Release Behavior:

set / export JBASE_DATE_FORMAT=1 for 'dd mm yyyy' (International)

set / export JBASE_DATE_FORMAT=2 for 'mm dd yyyy' (USA)

set / export JBASE_DATE_FORMAT=3 for 'yyyymmdd' (Japanese) (Only used with ICONV())

Any other value currently defaults to 'mm dd yyyy' (USA)

If JBASE_DATE_FORMAT is undefined then the Date Format is determined by JBASE_LOCALE (if set)

If JBASE_LOCALE is also undefined then use (Unix) 'locale' (LANG) / (Windows) 'Region and Language'

Patch Number: 5_20423

Ticket Number: 601468

Description: Q2Q pointer resolution fails when JEDIFILENAME_MD points to a location outside the account

Previous Release Behavior:

When the JEDIENABLEQ2Q environment variable is set, Q-pointer chains would not follow JEDIFILENAME_MD if it pointed to a MD not in the HOME directory.

Current Release Behavior:

LOGTO

Use the MD defined in the account's SYSTEM File entry.

If the MD is not defined in the SYSTEM File entry then use \$HOME/MD.

If no MD is available the use the default, i.e. \$JBCRELEASDIR/src/MDJD

Q2Q

If the Q-pointer points to another account, use the MD in that accounts SYSTEM File entry.

If the MD is not defined in the account's SYSTEM File then default to the MD in the account's \$HOME directory.

If no MD available anywhere, do nothing which results in a 'File not found' error.

Patch Number: 5_20424

Ticket Number: 600421

Description: Incorrect license info shown by 'jlicensinginfo'

Previous Release Behavior:

License info incorrectly displayed by 'jlicensinginfo'

Current Release Behavior:

License now correctly displayed by 'jlicensinginfo'