



jBASE Product Release Notice

Product:	jBASE 5.2
Version:	5.2.30
Type:	Patch Release

Contents

Features..... 3

Components 5

Supported Platforms 6

Patch Details..... 7

Installation Instructions for jBASE 5.2.30 8

Incidents Addressed in jBASE 5.2.30 Release 10

Patches Incorporated in jBASE 5.2.30 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 5.2.30 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 5.2.30. All Patch details for the jBASE 5.2.30 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

Installation Instructions for jBASE 5.2.30

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_5230.tar.gz
```

Install the jBASE release using:

```
tar -xvf AIX_53_JBASE_5230.tar
```


Windows Systems

Run the installer, e.g. 'WIN_JBASE_5230.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_pre5.2.30"

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

Incidents Addressed in jBASE 5.2.30 Release

Ticket Number	Description	Patch Number
566057	jAgent POODLE SSL Vulnerability	5_20398
564874	'SETPTR ?' does not display all information	5_20392
567563	License Validation utility	5_20391
567510	Issue when converting from UTF-8 back to Latin1	5_20389
568218	REFORMAT does not release handle on the source file	5_20394
568218	RENAME-FILE does not report an error when it fails	5_20390
557494	[JODBC] unable to connect to jBASE using ODBC and IBM Datastage	5_20399
572103	Update EULA and Copyright to Zumasys	N/A
563559	DREM and Secondary Indexes When there is an index on a DREM file, a SELECT against the DREM file fails but a KEY-SELECT works	5_20397
557512 \ 557508	Segmentation violation errors	5_20401
564879	Enhancement request for PH-KILL to set End Time and Date	5_20396
563619	EV command in JED issue	5_20388
564941	LIST-ITEM (S' does not suppress line numbers	5_20395
561147	jRFS processes increment license counts	5_20402
576276	jbase_agent does not return entire query, instead returns a subset result always in even thousands.	5_20403
576276	SELECT * aborts when parsing dictionaries	5_20404

Patches Incorporated in jBASE 5.2.30 Patch Release

Patch Number: 5_20388

Ticket Number: 563619

Description: The JED 'EV' command cannot be used on two records in the same file at the same time. Prior to this patch, the 2nd record would be locked and only be allowed READ ONLY access.

Patch Number: 5_20389

Ticket Number: 567510

Description: Update single attribute item using 'jutf8 -u' also prevents 'Segmentation Violation' when accessing 'non utf8 character sequence'

It appears that when you do the LATIN1 function on a UTF-8 string that contains code points that are not represented in ISO-8859-1, the conversion process is dropping the leading bytes up to the last byte in the UTF-8 sequence and then giving the result of the final byte hence the Euro character "€" which in UTF-8 is E282AC ends up dropping the E2 & 82 bytes and returning AC which equates back to the "-" character. You can demonstrate the same issue with any other code points over U+00FF.

Patch Number: 5_20390

Ticket Number: 568218

Description: Fix to return error message on RENAME-FILE command failure

Patch Number: 5_20391

Ticket Number: 567563

Description: License validation utility

In previous jBASE versions when you installed a license key, jBASE would effectively validate that key and report if the key was incorrect or couldn't be installed, but because the key installation in 5.2 is just inputting the details into the system.properties file, there isn't a similar mechanism for 5.2. jKeyCheck utility will display details on a licence key or interrogate the system.properties file.

Patch Number: 5_20392

Ticket Number: 564874

Description: 'SETPTR ?' does not display all information

'SETPTR ?' now displays what printer and banner were assigned and whether that SETPTR assignment included a HOLD option.

The behavior prior to this patch displayed:

```
jsh -->SETPTR ,,,,,,AT HP,HOLD
Unit Number :0
Page Width :132
Page Depth :66
Top Margin :3
Bottom Margin :3
Print mode : 1 - Output to HOLD file ( P#0000 )
```

```
jsh -->SETPTR ?
Unit Number :0
```

Page Width :132
Page Depth :66
Top Margin :3
Bottom Margin :3
Print mode : 1 - Spooled Output

The current behavior displays:

jsh -->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 : NONE

Initial Job State : PRINT

Patch Number: 5_20394

Ticket Number: 568218

Description: REFORMAT does not release handle on the source file (Windows only)

The following code would fail to rename the REF1 file.

```
001 CRT 'Reformatting REF1 -> REF2'  
002 DATA 'REF2'  
003 EXECUTE 'REFORMAT REF1 INV 1 2'  
004 CRT 'Renaming REF1 -> REF1INV'  
005 DATA 'REF1INV'  
006 EXECUTE 'RENAME-FILE REF1'
```

This was because REFORMAT had not closed REF1

Patch Number: 5_20395

Ticket Number: 564941

Description: 'LIST-ITEM (S)' does not suppress line numbers

Prior to this patch 'LIST-ITEM' and 'LIST-ITEM (S)' would produce the same output.

Patch Number: 5_20396

Ticket Number: 564879

Description: Enhance the PH-KILL command to set the End Date and End Time

PH-KILL command now collects END-DATE and END-TIME.

Patch Number: 5_20397

Ticket Number: 563559

Description: jEDI driver failed when the file has a secondary index

Prior to this patch, SELECT using a custom jEDI driver failed when the file has a secondary index. SELECT would fail by selecting a single item-id that did not exist on file.

Patch Number: 5_20398

Ticket Number: 566057

Description: jAgent POODLE SSL Vulnerability, disable SSL2/3

Once the POODLE SSL Vulnerability was exposed by Google many clients did not want to use SSL. This change will disable SSL in jbase_agent, and will always default to TLS1 for every connection.

Patch Number: 5_20399

Ticket Number: 557494

Description: jODBC unable to connect to jBASE using ODBC and IBM Datastage

Connecting from some ODBC applications will use a different mechanism than the more common Microsoft applications, clients were unable to connect via RUBY and IBM Datastage.

Patch Number: 5_20401

Ticket Number: 557512/557508

Description: Fix for segmentation violation

Clients experienced a memory leak when using dictionary a Tfile (i.e. Translate) conversion, also caused clients index queries to lock up. Check all threads and release any optional parameters passed via PERFORM/EXECUTE when closing down thread.

Patch Number: 5_20402

Ticket Number: 561147

Description: jRFS processes increment license counts

Any sessions that access a file via jRFS will have one license added for the command and one for the jRFS session. In 5.2.30, only one multisession should be allocated as the jRFS session is now ignored.

Patch Number: 5_20403

Ticket Number: 576276

Description: jbase_agent does not return entire query, instead returns a subset result always in even thousands

Fix exception caused by SQL library entering the debugger. Previously when SQL Selects used multiple ITYPE that called subroutines once the jBC debugger was entered it would cause an exception.

Patch Number: 5_20404

Ticket Number: 576276

Description: SELECT * aborts when parsing dictionaries

The default behavior to display a warning and ignore invalid dictionary items when performing select *. Selecting an invalid dictionary item, SELECT BAD_DICT, GOOD_DICT FROM MY_TABLE Will still fail, but now will also display a warning.