# Helix Client Classic 6.1.5 Release Notes

Helix Client

Classic 6.1.5

Release Notes

# 1 Welcome to Helix Client Classic 6.1.5

## 1.1 Introduction

Welcome to Helix Client Classic 6.1.5. This is the fifth maintenance release of the Helix 6.1 Client/Server product family. This release contains feature refinements and bug fixes that have been incorporated into the code.

## 1.2 About this document

This document is a supplement to *The Helix Reference*, the complete guide to Helix 4.5.5 and earlier. It is our intention to publish an up-to-date, self-contained Helix reference manual, but that manual is not available as of this writing.

This document describes the features, enhancements, and bug fixes made to Helix since the release of Helix Client Classic 6.0.x. This document is designed for people who are already familiar with Helix Client Classic 6.0.x and want to quickly learn about the changes made since that release.

This document covers all changes made in Helix Client Classic 6.1; fixes and enhancements made since the initial release are noted where appropriate. For a list of changes specific to each release, visit <http://www.qsatoolworks.com/product/helix61/client-server.html>

If you are not familiar with Helix Client Classic 6.0.x, you should also refer to the *Helix 6.0.1 Release Notes*, as that document covers changes made in earlier Helix 6 releases. You may also want to refer to the *Helix 5.3.2 Release Notes*, as that document covers changes made during the Helix 5 life-span.

QSA ToolWorks, LLC makes the information in this document available on an as is basis and is not responsible for its accuracy, use, or future compatibility with Helix products or other products such as the Macintosh® operating system.

## 1.3 About this release

Helix Client Classic 6.1.5 is a maintenance release, whose purpose is to provide access to collections served by Helix Server 6.1.5. Helix Client Classic 6.1.5 is (as the name implies) not Intel-native, but it is fully compatibly with both Helix Server for PowerPC 6.1.5 and Helix Server for Intel 6.1.5.

Helix Client Classic 6.1.5 is functionally equivalent to Helix Client Classic 6.1.4 — no new features or bug fixes have been introduced. Its sole purpose is to provide compatibility with Helix Server 6.1.5. The primary purpose of Helix Server 6.1.5 is to provide new features and bug fixes for the Mac OS X native Helix Client Preview Release. See the Helix Server 6.1.5 Release Notes for more information.

Helix Client/Server 6.1.5 is a general release and can be used with all existing collections. We encourage all customers to update existing collections to Helix Client/Server 6.1.5, regardless of whether or not they are experiencing problems in the areas that are specifically addressed by this release.

*Although we are confident that Helix Client/Server 6.1.5 is a solid release, it is still possible that problems may be found that could result in collection damage. QSA ToolWorks, LLC will do everything in its power to support users who encounter errors caused by this software, but we cannot guarantee that problems do not exist. As always, please be diligent regarding your backup and utility procedures.*

### 1.3.1 Mac OS X native progress

Helix Client Classic 6.1.5 represents an interim step in our work in updating our code for Mac OS X. Helix Server, Helix Utility and Update Collection are currently Mac OS X native. Mac OS X native versions of Helix Client and Engine are available in Preview Release versions.

It is our intention to release Mac OS X native products as soon as possible, but because of the technical issues that made it impossible to reliably run Helix Server in Classic Mode in Mac OS X, we chose to focus first on the Mac OS X native Helix Server and to release it as soon as it was ready. That task was completed in December, 2005.

Shortly thereafter, Apple announced the transition from the PowerPC processor to Intel processors. That transition was completed in 2007. Unfortunately, Intel-based Macs do not support the Classic environment, and writing programs that run natively on them requires a switch to Xcode, Apple's proprietary programming environment. In addition, Mac OS X 10.5 (aka: Leopard) which was released in October, 2007 does not support the Classic environment on PowerPC-based Macs. These events created a pressing need for Mac OS X native versions of all Helix products.

The technical nature of the shared components within the Helix product line make it most logical for us to deliver Mac OS X native versions of Helix Client and Engine next. Helix RADE will be the last product to become Mac OS X native.

The Mac OS X Transition Journal on our web site contains the latest updates on this process.

### 1.3.2 Mac OS X native Helix Client Preview Release

*Helix Client Classic 6.1.5 is the only officially released Helix Client at this time. It is fully compatible with both Helix Server for PowerPC and Helix Server for Intel. The information in this section is provided for those who want a preview of the Mac OS X native Helix Client.*

A Mac OS X native Helix Client 6.1.5 — for both PowerPC and Intel — is available as a Preview Release via the Preview Release section of our website: <http://www.qsatoolworks.com/product/preview/>. Helix Server 6.1.5 is required to use these Preview Release products.

*Preview Release versions are incomplete versions of our full products. They may do everything you need Helix to do, in which case they should work fine for you. If they do not, check the 'Non-Functional' list (in the **About Helix Client** window) to see if the function you need is listed there. If something does not work for you and the problem is not noted on the Preview Release web site, please file a bug report using the instructions found on the site. (Telephone and email bug reports for Preview Release products are not accepted.)*

It is our intention to deliver subsequent Preview Release versions of Helix Client on a regular basis, as we add features and fix reported bugs. The new **About Helix Client** window contains a live link to our web site, where information on the latest version of Helix Client is shown. We recommend that you check on a regular basis to see if an updated version of Helix Client is available.

Because of the rapidly changing nature of the OS X native Helix Client for Helix Client Preview Releases, information about them is not contained in these Release Notes. Please check the Preview Release web site: <http://www.qsatoolworks.com/product/preview/> for documentation on the Helix Client Preview Release.

# 2 Contacting QSA ToolWorks, LLC

## 2.1 Sales and customer service

If you need to purchase new or upgraded Helix products, you can visit our online store at <http://store.qsatoolworks.com>. You can also contact our sales and customer service department by email <sales@qsatoolworks.com> or by phone 800-784-7018. General questions and other administrative issues should also be directed to the customer service department. The sales office is open 9AM–5PM Eastern Time (1400–2200 GMT).

## 2.2 Technical support

If you encounter technical problems (or have general technical questions) related to Helix, technical support is here to help. You can contact technical support by email <support@qsatoolworks.com> or by phone 570-662-8883. Telephone support is available 9AM–5PM Eastern Time (1400–2200 GMT). Some support services are subject to additional fees.

The Technical Support section of our web site <http://www.qsatoolworks.com> contains the latest information about the support services we offer.

### 2.2.1 techdb: the Helix-based source for technical support

techdb is a Helix Client/Server database that we use to provide technical support to our customers. In techdb you can search our knowledgebase, look up Helix error codes, submit bug reports and feature requests, chat with other Helix users, and more.

In essence, techdb is a live demo of Helix Client/Server in action, and it provides a communal gathering spot for Helix users around the world. That alone makes it a worthwhile part of our technical support services.

Instructions on accessing techdb are found at…
<http://www.qsatoolworks.com/support/techdb.html>

### 2.2.2 Sending files

Please do not send us files via email. Because of the increasing problem of malware sent as email attachments, unsolicited attachments are automatically deleted. If you have a file you wish to send to us, please contact us first and describe the problem. We may have already isolated the source of the problem and can discuss possible remedies without needing to see your example. If we do need to see the file you have, we will send instructions so you can send it in a way that will ensure the fastest possible action on our part.

### 2.2.3 About collection repair

Collection repair is no longer charged as a flat fee service. Repairs are now billed on an hourly basis, as per our User Support Unit (USU) policy. Many repairs can be covered with the USUs that come with each Helix upgrade, effectively costing nothing extra. We strive to turn all repairs around as quickly as possible; typical turnaround time is under 8 hours. Please see the support section of our web site for more details.

## 2.3 Bug Reports

If we are to continue to improve the product and meet the needs of our customers, we need to know when you experience problems. Bug reports can now be submitted via the interactive bug reporting section of techdb. See section 2.2.1 (*techdb: the Helix-based source for technical support*) for information on connecting to techdb.

## 2.4 Feature requests

If you wish to submit a request for a feature you would like to see added to Helix in the future, please visit the feature request page on our website and tell us about it. This is an automated system and you will not receive a personal reply from QSA ToolWorks, LLC to items submitted there.

We hope to integrate Feature Requests into techdb soon.

# 3 Configuring Your Computer

## 3.1 Introduction

The following information is designed to help you achieve maximum performance with Helix. Other applications that you use may prefer different settings. None of these recommendations are required for Helix to work properly, but if you discover a problem while using this version of Helix, please confirm that it also occurs with these settings in effect. If you discover something that behaves differently based on your Mac OS configuration, please inform us of the details.

## 3.2 Mac OS 9.1 or higher required

Helix Client Classic 6.1 requires Mac OS 9.1 or higher. Helix Client Classic 6.1 also runs in Classic mode on PowerPC based Macs running Mac OS X 10.1 through 10.4.x.)

## 3.3 Installing Helix Client Classic 6.1

Installing Helix Client is simply a matter of downloading the software from our web site (keeping in mind which version of Helix Server you will be connecting to) and running the installer package. The installer asks you to choose the folder into which Helix Client Classic 6.1 will be placed. We recommend installing into the **Applications (OS 9)** folder.

### 3.3.1 Enabling Helix Client Classic 6.1

After installation, you must *enable* your copy of Helix Client Classic 6.1. To enable Helix Client, simply launch the program and supply your personal registration information (name, address, etc.) and click the OK button.

# 4 What's New in Helix Client Classic 6.1

*Helix Client Classic 6.1.5 is functionally equivalent to Helix Client Classic 6.1.4 & 6.1.3 — no new features or bug fixes have been introduced. Its purpose is to provide compatibility with Helix Server 6.1.5.*

*The changes listed below were introduced in prior Helix 6.1.x releases.*

## 4.1 New Features in Helix Client/Server 6.1.3

*The following features are new in Helix Client/Server 6.1.3.*

### 4.1.1 OS X text files can now be displayed

When Helix displays the contents of a document, the Picture tile first checks the Classic TYPE (an internal code) for TEXT and EPSF document types (which are handled internally) and then passes all other types to QuickTime for conversion. Because Mac OS X does not use the type and creator codes many Mac OS X text documents could not be displayed via the Picture tile. Helix Server now completes the loop: if QuickTime returns the 'document format unsupported' error, Helix Client now displays the file as plain text. This allows all sorts of text files (e.g. crash logs) to be displayed as plain text. However, other non-graphic file types (e.g. .dmg) are also shown as plain text, which can be disconcerting to a non-technical end user. Nonetheless, this ability can be useful when diagnosing problems, so it has been left in.

## 4.2 Keyboard shortcut modifiers supported

In addition to the original Command key commands, Shift-Command, Option-Command, and Shift-Option-Command modifiers are now available. Contact your collection designer to have extended modifiers added to your Helix Client user menus.

## 4.3 Keyword indexing and the HKWT resource

For collections that contain fields with keyword indexes, the characters that constitute word separators are contained in the HKWT (Keyword Separator Table) resource. If your use of Helix Client entails the use of keyword-based searches, contact your collection designer to find out how these changes affect you.

### 4.3.1 Changes to the Keyword Separator Table

The following information is of interest only when collections use keyword-based searches.

#### 4.3.1.1 Missing characters added

In prior versions of Helix, many 'High ASCII' characters that are used frequently in European languages were being ignored as word characters. Consequently, words containing, characters such as Å and Ø were excluded from keyword searches.

#### 4.3.1.2 Ligatures added

The two ligatures included in the standard Macintosh font set — fi & fl — are now included in the word character set.

#### 4.3.1.3 Non-breaking space added

Previously, the non-breaking space (NBSP) character was not considered a word character before, even thought the very definition of the non-breaking space speaks for its inclusion.

Helix 6.1 treats a non-breaking space — created by typing option-space — as a word character.

### 4.3.2 Inaccurate information

Previously published information, such as that found in Appendix A of *The Helix Reference* is inaccurate. For example: É is noted as being a word character, but in the actual HKWT resource, it was being treated as a separator. Many similar inconsistencies were found and corrected.

#### 4.3.2.1 Updated and corrected tables available via technote

A Helix Keyword Separator Table technote containing up-to-date information is available at: <http://www.qsatoolworks.com/support/technical/knowledgebase/kbfiles/technotes/hkwt/>

Tables showing the correct information for the HKWT resource in prior versions of Helix (replacing Appendix A of *The Helix Reference*) — as well as a table showing the new HKWT resource and a comparison chart showing the differences — are found in the technote referenced above.

# 5 Bug Fixes

*Helix Client Classic 6.1.5 is functionally equivalent to Helix Client Classic 6.1.3 & 6.1.4 — no new features or bug fixes have been introduced. Its purpose is to provide compatibility with Helix Server 6.1.5.*

*The changes listed below were introduced in prior Helix 6.1.x releases.*

## 5.1 Bugs Fixed in Helix Client/Server 6.1.3

*The following bug fixes are new in Helix Client/Server 6.1.3.*

### 5.1.1 Post on Print could cause Helix Server crashes

Many bugs that could cause Helix Server crashes when a Helix Client prints a view that has posting icons attached to the On Print column have been fixed. Posting while printing should be much more stable.

### 5.1.2 External document displayed as 'format unsupported' (BZ957)

A bug where Helix Client Classic would see a 'format unsupported' message when Helix Server for Intel attempted to render a picture from an externally stored document is fixed.

### 5.1.3 Reports with over 32,766 records failed to display all records (BZ976)

A bug that caused Helix Client Classic (and RADE) to fail to show all of the records in a report (list) when there were more than 32,766 records in the report is fixed.

### 5.1.4 Fixed Point Posting Trigger Multiplied by 100 (BZ1034)

A bug that caused conditional sequences to run 100 times the expected value when a fixed point value is used as the trigger *and* that same value is posted to another relation, has been fixed.

### 5.1.5 Eszett (ß) character (0xA7) is a word character in German

When the keyword separator table was updated, the German eszett character was inadvertently left out of the word character list.

*Helix customers who use the eszett character in text stored in keyword fields need to rebuild all keyword indexes or keyword search results will be inaccurate.*

See "Keyword indexing and the HKWT resource" on page 5 for more information about keyword indexes.

### 5.1.6 Collection Damage reported after Apple Events access (BZ979)

Helix 6.1 does not use the 'compiled form' data that was used in Helix 6.0 and earlier, but Update Collection is not aware of this. In cases where Apple Event access users neglected to close a process, this obsolete data would not be cleared properly. Because Update Collection is backward compatible with Helix 6.0, it considers this a problem and reports it as collection damage. In reality there is no damage; Update Collection is being overly sensitive.

Helix 6.1.3 takes greater care to discard this obsolete data, keeping Update Collection from falsely reporting damage.

### 5.1.7 Registration Dialog Issues (BZ904)

The registration dialog would not allow the user to switch to another program, copy data, switch back and paste that new data into a field. This is fixed.

## 5.2 Bugs Fixed in Helix Client/Server 6.1.2

*The following bug fixes are new in Helix Client/Server 6.1.2.*

### 5.2.1 Intel-native only

#### 5.2.1.1 Queries on dates failed

A date typed into a query was being stored incorrectly in Intel-native Helix. The result was queries that would return either all records or no records, regardless of the date stored in the field. This bug has been fixed.

This bug affected all types of queries (quick, form, and power) where a date was literally typed into the query, as opposed to a query where the date is derived from data stored in a record.

#### 5.2.1.2 Helix Server crash when handling styled text

When a Helix Client caused a change to occur to a field containing styled text, Helix Server could scramble some important memory locations, resulting in a Helix Server crash. This bug has been fixed.

### 5.2.2 Pasting large pictures failed

Pasting data larger than 32KB into a Picture field failed to store the picture properly. This was immediately recognizable, as the contents of the rectangle would not match the picture being pasted into it, typically appearing as a blank. This bug was non-destructive and other than the fact that the pasted picture was not properly saved, there was no negative impact.

This bug has been fixed. Helix now handles pasted picture data correctly regardless of size.

### 5.2.3 Helix Server stability improved

Changes to some core routines in Helix Server 6.1 result in improved stability for both the PowerPC and Intel versions of Helix Server 6.1.2.

## 5.3 Bugs Fixed in Helix Client/Server 6.1.1

*The following bug fixes were addressed in Helix Client Classic 6.1.1.*

### 5.3.1 Document corruption

A bug in Helix Client/Server 6.1 would sporadically result in internally stored documents being corrupted when the **Copy From Volume** command was used to upload a document.

The most insidious part of this bug was that there was no warning that a document had become corrupted during the upload. Only when retrieving a document at a later date would Helix Client report that it "detected a problem with the document" and offer to delete the download or let you try again. Trying again would not help, as the corruption was actually in the stored document.

Helix Client/Server 6.1.1 fixes this bug, and we recommend that all customers who use internal documents install this update immediately.

In addition to fixing this bug, the new code in Helix Server computes the document checksum during more stages of the document transfer, resulting in an more robust transfer mechanism that will report other types of network problems that may have gone undetected in the past.

*If you store documents internally in your collection, you should increase the preferred size of Helix Client so it is large enough to hold the entire document in memory while uploading it. A good rule of thumb is to estimate the size of your largest document (in KB) and add 10000 KB to that number. For example, if your documents are as large as 14 MB (14000 KB) then set your Helix Client Classic 6.1's Preferred Size to 24000 KB.*

*To increase the preferred memory: highlight the Helix Client Classic 6.1 icon in the Finder, and choose the **Get Info** command. Then open the **Memory** section and change the value in the **Preferred Size** field. (The **Minimum Size** field does not need to be changed.)*

### 5.3.2 Helix Client non-responsive when Helix Server dialogs are open

When a user chooses **Quit** on a Helix Server that has Clients connected, three dialog boxes are presented. The first is the standard "Save changes before Quitting?" dialog; the second informs the user that there are Clients connected, giving them one last opportunity to cancel the Quit

operation. The third dialog states that the Server will quit when all Clients have logged off and offers only an **Abandon** button.

In prior versions of Helix Server, while either of the first two dialogs were open, connected Clients were unable to communicate with Server, making it impossible for them to finish their work or even to disconnect from the Server. Both of those dialogs now allow Client tasks to continue even while they are open on the Server's screen.

Nonetheless, it is still recommended that these dialogs be answered immediately. Network performance is significantly reduced when these dialogs are left open.

### 5.3.3 AutoOpen Post with omitted key field corrupts index (BZ777)

A rare combination of factors would result in the corruption of an index, resulting in records appearing twice in lists, or deleted records appearing as a blank line in lists. This has been fixed.

Helix Utility has always detected this problem and scheduled the index for rebuilding, but lists could display inaccurate data between Helix Utility runs.

### 5.3.4 Crash Recovery: pictures over 64KB are lost (TS2233)

Pictures over 64KB, pasted into Picture fields, were not being recovered when the log file was applied after a crash. Upon recovery, the records containing the pictures were recovered, but the picture field was left empty.

This is now fixed. All data is now properly recovered from the log file after a crash.

### 5.3.5 Potential crash on export

During export operations an internal counter was set too high. This would occasionally cause a crash when Helix was running natively in Mac OS X. This bug has existed since at least Helix Express 4.0 (and probably since Helix 1.0) and although we never had a crash report in Helix Client/Server directly attributable to this bug, it was a theoretical possibility. Therefore the fix was also applied to Helix Client Classic 6.1.1.

### 5.3.6 Potential write lock failure

It was discovered that every four billionth attempt to write lock a record would fail, and the record could potentially be edited by two users simultaneously. A user would have to write lock a record once every second of every day for 136 years to reach this number, so we are confident this bug has never affected any Helix user to date. Nonetheless, it is now fixed and will not affect users no matter how many centuries their databases are kept in use.

### 5.3.7 'Mini menu' cleaned up

The items available in the 'mini menu' — the one that appears when you launch Helix Client but have not yet connected to a Helix Server — contained a number of pointless commands that were always disabled. Those have been removed.

### 5.3.8 Miscellaneous fixes

Because the main focus of our effort since the release of Helix Client/Server 6.1 has been on completing the Intel-native version of Helix Server and advancing the Mac OS X Helix Clients, many changes have been made to the communications core that exchanges data between Helix Server and Helix Clients. Although not cataloged specifically as Helix Client Classic 6.1.1 fixes, we expect that many of these changes will have a beneficial effect in Helix Client Classic 6.1.1 as well.

## 5.4 Bugs Fixed in Helix Client/Server 6.1

*The following bug fixes first appeared in Helix Client/Server 6.1.*

### 5.4.1 Intermittent posting failures (TS2013)

In rare situations, posting mysteriously stops working after a view is in use for a while. When this bug strikes, the post stops working and does not work again until remedial action is taken.

This bug is extremely rare and only seems to affect relations where there are a large number of posts involved. The bug actually affects the internal posting package that is created when the posting is set up by the collection designer.

This bug has been fixed by doing away with these packages altogether in Helix 6.1. The performance gain on modern hardware was negligible, and the added complexity was allowing subtle bugs like this to occur.

### 5.4.2 User view permissions inadvertently restricted (TS1992)

An obscure bug in the Helix RADE User Editor has been fixed. This bug could cause inadvertent restrictions on views when the collection is used with Helix Client.

If your Helix Client user has restricted permissions where it should not, contact your collection designer to apply this fix.

### 5.4.3 Enter menu highlighting

Pressing a command key to execute a menu item should always highlight that menu briefly to provide feedback to the user. In Helix 4.5.3 this function was disabled in the case of the Enter command. The rationale was that when Enter is called repeatedly in a sequence, highlighting the menu is a distraction and reduces performance.

In Helix Client Classic 6.1, pressing the Enter key properly highlights the menu when an Enter command is executed. When the Enter command is executed as part of a sequence, the menu is not highlighted.

### 5.4.4 Why? message during sequence: format error

When an error occurs during the running of a sequence, there can sometimes be two error messages generated. For example, if a sequence attempts to enter a record that can not be entered because a validation check fails, there are two errors involved: the error because of the validation failure and the error because the record can not be entered.

Prior versions of Helix concatenated these two errors into a single entity, sometimes producing messages that did not make sense as presented. Helix 6.1 splits these messages onto separate lines, making it more apparent that two errors are involved.

### 5.4.5 Build number missing from splash screen

Prior versions of Helix Client were not displaying the build number in the splash screen, making it difficult to determine precisely which version of Helix was in use. The build number is now shown when Helix Client launches and when selecting About Helix Client from the Apple menu.

# 6 Known Problems

This list of known problems was compiled from previous release notes as well as recent additions made to our web site Knowledgebase. Check <http://www.qsatoolworks.com/support/technical/knowledgebase/> for the latest updates to the known problem list.

## 6.1 Known problems new to Helix 6.1

### 6.1.1 Performance degradation on WAN

Although Helix Client/Server 6.1 users on a local area network (LAN) will find performance to be equal or better than Helix 6.0, Helix Clients accessing a Server across a wide area network (WAN) may find performance not as good as it was in Helix Client Classic 6.0.x. Changes required to create the Intel-native products require us to generate smaller data packets, which are delayed by the latencies involved in routing TCP/IP traffic. We continue to work on methods of improving the performance of Helix Client/Server for both LAN and WAN connections.

## 6.2 Pre-existing known problems

### 6.2.1 Conditional Sequence can run too many times (BZ1034)

When a fixed point abacus is used as a source value in a post (either in the destination tile or a posting triplet) that same abacus is used to specify the On Condition value of an optional (i.e: conditional) sequence, *and* that abacus is not present on the template, the conditional sequence runs 100 times as often as specified.

### 6.2.2 Fragile indexes are not invalidated

If a Client changes data that requires the rebuilding of a fragile index, the index is not invalidated by Helix Server. The Helix Utility Data Damage Repair process will flag this index, causing this index to rebuild the next time the collection is opened.

Index results will be inaccurate until the index is rebuilt.

### 6.2.3 Printing issues

#### 6.2.3.1 Record data reverts when printed if post on print applies to the view

If an existing record is displayed and a text field is modified, the record can be printed with the altered (but not entered) text. Subsequently closing the record appropriately asks the user whether they want to save the changes or not.

If a post set to trigger “on print” is attached to the view, the record is printed with the altered text, but the record is immediately reverted to its original state after printing, and the user is not given an opportunity to save the changes.

If you are experiencing this problem, ask your collection designer to apply the fix as outlined in the Helix RADE Release Notes.

#### 6.2.3.2 Find and Print All with more than 128 records

Using **Find and Print All** to print more than 128 records with ‘n-up’ layout only prints the first 128 records in n-up form. Subsequent pages print with the 1-up setting. For example, printing 150 records with 4-up layout results in the first 128 records printing (at the properly reduced size) on 32 pages, followed by records 129–150 printing one per page.

To work around this problem, select your desired n-up setting in the print dialog, and then click the “Save Settings” button. This will tell the LaserWriter driver to use that setting for the entire print job. Remember to reset the print dialog to 1-up layout when you want to print records at full-size.

#### 6.2.3.3 Last visible item on a list view fails to print

The last item on a list view can fail to print even though it displays properly on screen. The problem is that the last time the “Page Setup” command was used on the view, a different printer (or different printer page size) was selected. When page sizes are changed, Apple’s printer drivers display a dialog that remind the user to “be sure to choose Page Setup in all open applications” and avoid printing issues. Because of Helix’s ability to remember page set-

tings for playback in sequences, it always retains the previous Page Setup information. During printing, Helix first calculates how many records fit on each page. Unfortunately it uses the stored Page Setup information to do this calculation. If the printable area becomes slightly larger, as is common with newer printers and their smaller page margins, Helix's old calculations are incorrect, and one or more records is trimmed off.

To solve the problem, choose Page Setup and click "OK" — all displayed records will now print.

### 6.2.4 Inert field behavior

In certain instances (e.g. posting) the value contained in an inert field is only recognized after tabbing out of it. Even if the value showing in the data rectangle appears to be changed, processes depending on the changed value do not occur unless the inert field is tabbed through.

#### 6.2.4.1 "The Enter key cannot be pressed here"

If the first data rectangle on a view contains an inert field, Helix may refuse attempts to enter a record with a Why? message indicating that "The enter key cannot be pressed here." There are a number of known workarounds, among them…

- When Helix refuses to enter the record, open the Why? message and dismiss it. The record can now be entered.
- When Helix refuses to enter the record, close the view. Helix will display the standard "Enter, Discard, or Cancel?" dialog. Clicking the Enter button stores the record. Note that this only works if the User Mode user has add permission for this view.
- Ask your collection designer to apply another workaround outlined in the Helix RADE Release Notes.

### 6.2.5 Option 4 posting in Find and Print All (BZ740)

A bug has been discovered that occurs during record printing if and only if you have all of seven distinct conditions present. In this situation, the records print properly until the first multi-paged record is encountered. After that record is printed, subsequent host records are deleted before they are printed, and their pages are printed with no data.

*This bug has existed in Helix for as far back as we can tell, but it was not discovered until the Helix 5.2 test cycle. If you have never encountered it before, you do not need to worry that your collection will change behavior due to a Helix version upgrade.*

This particular set of conditions is very rare, but if you believe you have experienced this problem, ask your collection designer to apply one of the workarounds outlined in the Helix RADE Release Notes.

### 6.2.6 Deleted field reverts to its previous value after entry (TS958)

A field with a default abacus will not store a default value if…

1. The record was previously entered with a value in this field, and
2. You simply delete the stored value and tab out, letting the default 'drop in' to the field.

*This bug has existed since the release of Helix Express 3.0. If you have never encountered it before, you do not need to worry that your collection will change behavior due to a Helix version upgrade.*

If you believe you have experienced this problem, ask your collection designer to apply one of the workarounds outlined in the Helix RADE Release Notes.

### 6.2.7 Power Query text edit field issues

The text edit field in a power query window does not properly respond to mouse actions such as clicking or dragging.

*This bug has existed since the Power Query was introduced.*

**Workaround**: Arrow key editing functions do work as expected.

### 6.2.8 Dynamic popups limited to 254 items

Dynamic popup menus are limited to 254 items. Popup lists with more than 254 items will display the non-functional text “More…” as the 255th item. This item does nothing. If the relation from which the dynamic data is drawn contains more than 254 items, you must use a query to limit the popup’s content.

*This limitation is removed in the Mac OS X native Helix Client.*

### 6.2.9 Subform (or list) won’t open the selected entry view (TS960)

A list view with an attached entry view does not respond when attempting to open a record. When you double click a record, one of the following is observed:

- As a subform: the cursor does not change to the hand symbol and double clicking does not open the entry view.
- As a list: the cursor changes to the hand symbol and the record highlights, but double clicking does not open the entry view.

If you are experiencing this problem, contact your collection designer to apply the fix outlined in the Helix RADE Release Notes.

### 6.2.10 Page number overflow

When the “P# is Page Number” feature is used on a view that has more than 65,535 pages in it, or on a list that has more than 65,535 records in it, the counter will overflow and begin counting up from 0 again on the 65,536th element.

### 6.2.11 Text field overflow

Attempting to type more than 32,767 characters into a field will result in a Why? message stating the field cannot contain more than 32,767 characters. Other methods of adding characters to a field (e.g: importing, posting, Apple Events) can bypass this check, resulting in a crash.

### 6.2.12 Large list record selection

Although Helix can view and scroll through lists containing millions of records without difficulty, you may still have problems viewing and scrolling through larger lists. However, Helix Client cannot properly select more than 16,382 records in a list.

Using Shift-click or Command-click to select more than 16,382 records results in a dialog box showing the total record count you originally selected, but noting that “The list has been truncated to 16,382 records.” The actual number of records selected is indeed 16,382, beginning with the first record you selected. Continued attempts to select more records via Shift-click or Command-click results in the selected record count being reported as 16,382, and only the original 16,382 records are actually selected. A subsequent Copy Selected Records copies only the highlighted records to the clipboard.

Choosing Select All Records in a list with more than 16,382 records results in a dialog box stating the total (correct) record count in the list but noting that only 16,382 records have been selected. In fact, *all records are selected* even though only the first 16,382 records are highlighted. A subsequent Copy Selected Records, Paste Selected Records, or Delete Selected Records acts on all of the records in the list, not just the highlighted records.

### 6.2.13 Document handling

#### 6.2.13.1 Importing external documents

When importing into Helix, an external document path will result in an error if the file that the path resolves to can not be found. Because of the variety of ways that Helix is used to import and export document references, this problem requires further study before a solution can be proposed.

#### 6.2.13.2 External documents in text and EPS formats not displayed

The text and EPS document display code was originally custom written for use with Helix’s internal documents. Neither of these document types are supported by the QuickTime conversion engine, so they are not viewable when stored externally. To display the contents of text and EPS documents, the documents must be stored internally.

### 6.2.14 Use of Helix Custom Help

Helix versions 4.5.3 and beyond do not officially support the old Helix Help or Helix Custom Help files. You can use old files if you insert the registered trademark symbol (®) after the word Helix (e.g. Helix® Custom Help).

### 6.2.15 Sleep mode crash and background non-responsiveness

#### 6.2.15.1 Sleep mode can cause Helix Client crash

Helix Client does not get along well with Sleep Mode. (Monitor sleep is not a problem.) Collections opened with Helix Client should be closed if there is a chance that you will be away long enough to have your system enter the sleep mode.

Users who run in Mac OS X Classic Mode should set their Classic Preferences Panel (Advanced Tab) to never allow Classic to sleep. They should also be conscientious about quitting Helix Client when it is not in use.

#### 6.2.15.2 Helix Client in the background becomes non-responsive

In addition, users of Helix Client Classic 6.1 have reported an increase of incidents where Helix Client becomes non-responsive after being placed in the background when running in Mac OS X Classic Mode. To avoid this problem, be sure to quit Helix Client instead of putting it into the background if it is not going to be used for an extended time.

### 6.2.16 Cosmetic problems

#### 6.2.16.1 Proportional scroll boxes and asynchronous views

Because Helix draws views asynchronously (that is, it draws the view first, then dispatches a separate routine to draw the data while it carries on with other tasks) it is impossible to properly set the vertical proportional scroll box of a view when it is opened. After the view has filled in the data, the scroll box may or may not recalculate and display with correct proportionality.

**Workaround**: Deactivating and reactivating the view, or simply resizing it slightly, will enable Helix to properly calculate the proportional scroll box.

#### 6.2.16.2 Import/Export dialog options command key

In import and export dialogs, CMD-O is the keyboard equivalent for the “Options” button. If you select a folder in the directory list, the “Import” or “Export” button changes to read “Open” as it should. Pressing CMD-O while the button reads “Open” flashes both the Options and the Open buttons, even though only the Options command is executed. This is a cosmetic problem only.

#### 6.2.16.3 Delete data dialog when subforms are involved

The delete data confirmation dialog that appears when a form with subforms is used does not center over the parent window as it should.

#### 6.2.16.4 Appearance manager compliance

The “Standard File” dialogs (Open collection, Get Document, etc.) can not be made appearance compliant. They will be replaced with Navigation Services style dialogs in the Mac OS X release.

# 7 Functional Clarifications

## 7.1 Two-digit date input

Helix versions 4.5.1 and earlier always interpret a two digit year input as being in the 1900s. Helix 4.5.3 and later use an HY2K resource to determine the beginning of a 100 year range which is used to interpret a user's two digit year input. The default value for this resource is 1920, which sets the interpreted range to 1920–2019. If you enter a year as a two digit number, Helix will interpret the date according to these rules.

If you prefer a different default date range, contact your collection designer.

## 7.2 'Previous' tile behavior when undefined fields are encountered

If a 'Previous' tile (Previous, Form Previous, and Previous For) encounters a record whose target field is undefined, it continues to search back (even through multiple records) until it finds a defined value to return. Two previous tiles on the same view, displaying the 'previous' value of two different fields, will actually display data from *two different records* if one of the target fields is undefined in the actual record that precedes the displayed record.

*This behavior has been consistent since at least Double Helix 3.5r13 (and probably earlier).*

If this causes a problem for you, contact your collection designer to apply one of the workarounds as outlined in the Helix RADE Release Notes.

## 7.3 Index limitations

Indexes are limited in the length of the significant data. Each additional level of sorting reduces the total length of significant data. The exact length of the significant data is dependent on the types of data in the index. You cannot rely on precise sorting if the total length of the data for all fields and abaci in the index exceeds 200 characters. Each additional sorting level further reduces the length of significant data by approximately 4 characters.

## 7.4 Rounding in Helix

Rounding is generally a pretty straightforward topic: when you round, round to the nearest number for the level of precision specified. Confusion is introduced when the value to be rounded is exactly halfway between the two possible values. This section is designed to explain and document how Helix rounds those values.

*For the sake of illustration, we will deal with rounding to the nearest integer, typically done with Helix's "Round" tile. Keep in mind that these same rules apply to the "Round to Nearest" tile, which allows you to specify the rounding precision.*

Rounding always seeks the nearest value to round to, but when the original value is exactly halfway between the target values, Helix employs one of two distinct sets of rounding rules. The rounding rules used depend on the data type being rounded.

The rules below only apply to values that fall exactly halfway between the two possible values. All other values round to the nearest number.

### 7.4.1 Rounding fixed points: financial rounding

Fixed Point data (technically, *integers*) were introduced much later in Helix's history and have always used the financial rounding method. Financial rounding seeks to make rounding more predictable, so it rounds *away from zero*, thereby guaranteeing consistent rounding.

*Financial rounding is also known as "symmetric arithmetic rounding," "banker's rounding," and a few others names. There are also variations on the basic rules for financial rounding — always rounding toward zero is one common variation — so be aware that not everybody has the same rules in mind when speaking about financial rounding.*

To see financial rounding in another setting, open Apple's ScriptEditor and run this AppleScript (make sure the event log is open so you can see the results).

```
repeat with i from -10 to 10
  get round (i + 0.5) rounding as taught in school
end repeat
```

In AppleScript, `as taught in school` uses financial rounding, and you will get the exact answers that Helix produces when rounding data in fixed point format.

## 7.4.2 Rounding numbers: scientific rounding

Number data (technically, *floating point numbers*) were introduced in the very first version of Helix and have always used the scientific rounding method. Scientific rounding seeks to minimize cumulative errors, so it rounds to the nearest *even* number, thereby reducing the possibility that the result will be skewed.

To see scientific rounding in another setting, open Apple's ScriptEditor and run this simple AppleScript (make sure the event log is open so you can see the results).

```
repeat with i from -10 to 10
  get round (i + 0.5) rounding to nearest
end repeat
```

In AppleScript, `to nearest` uses scientific rounding, giving you the exact answers Helix produces when rounding data in number format.

### 7.4.2.1 Decimal to binary conversion errors

Certain numbers can be seen to violate the specification for scientific rounding. This problem is introduced because computers typically convert decimal (as in base 10) numbers into their binary (base 2) equivalents before doing mathematical operations. In decimal math, we have many fractional values (e.g. 1/3) that we understand to be 'infinitely repeating decimal numbers.' Attempting to divide 100 identical items evenly between three people is impossible. Doing mathematical operations along this line introduces rounding errors that we naturally understand and compensate for. A human being can look at ((100/3)*3) and understand that the answer is 100, but if you work it out, the answer comes out as 99.999… and we simply 'round it off' to 100. But it is important to keep in mind that 100 is an approximation, the 'real' answer is 99.999…

When examining binary numbers, you find that a whole different series of fractional numbers turn out to be infinitely repeating. For example 1/10 is an infinitely repeating binary number.

Now consider how this applies to rounding. Given the number 0.235 and being asked to round to the nearest 0.01, you would apply the financial rounding rules and arrive at the (correct) answer of 0.24. However, Helix rounds this to 0.23.

Why? The answer is binary conversion error. 235/1000 is, when expressed as a binary number, an infinitely repeating number. Converting 235/1000 to binary and then back to decimal will yield (approximately) 0.23499999… Because *this* value is not exactly halfway between the two numbers we are potentially rounding to, it is rounded to the nearest value (down, in this case) and the result appears incorrect.

Why does Helix make this mistake? The math routines that Helix uses are part of the Macintosh CPU's ROM: they are part of the common package that most programs use. Open your Classic Calculator DA and type `1-.9-.1=` and you'll see that the answer is `2.032879E-20`, *not* `0`. Remember that 1/10 (0.1 in decimal notation) can not be accurately represented in binary: it is an infinitely repeating value in binary. The value has to be approximated, and the minute error is seen when math operations are performed.

If you do not need more than two decimal places of precision, you can avoid these rounding errors by converting the number into a Fixed Point data type before performing math operations, converting the result back to a number data type if necessary.