Active4D v6.3r2

IMPORTANT!
The minimum version of 4D for use with Active4D is v11.5 (including betas and master candidates). On Mac OS X, the minimum system version is 10.6 (Snow Leopard). The shell and demo databases are made with 4D v13.

NOTE: Active4D v6.3 is a paid upgrade for all users whose licenses are current, unless you purchased a v6.2 upgrade or original license after 12 Sep 2015.

Documentation

In addition to a complete documentation set specific to v6, there is also an Addendum that covers changes to v6 relative to v4.x. If you are upgrading from v4.x or v5, please read it carefully so you will know what has changed and what issues you may run into.

Requirements

Software:

4D v11.5 or later.

Hardware:

Mac OS X 10.5+, Intel processor
Windows XP SP2+

Upgrading to v6.x

There are important changes to the shell. Be sure to copy the methods **A4D_Init** and **A4D_OnWebConnection** from the latest shell database to your database.

Known issues

- **SET QUERY LIMIT** is not obeyed when used with **QUERY BY FORMULA**. This is a bug in 4D.

- As of 4D v11.5, when you use one of the record selection commands (such as **goto selected record**) along with the auto relate many option, *and* the one to many relation you are interested in is manual, the related many records will **not** be selected. Currently the workaround is to explicitly use **relate many**.

- Active4D defines all of the 4D 2004 and 4D v11 named constants, but no longer has access to custom constants that came from 4DK# resources. However, you can easily convert your custom constants to an Active4D library that will give you access to them. Please see “convert_constants.pdf” for information.

- Do **not** attempt to save a collection or session in v6 to a blob or document and then load it in v4.5. Saved collections are backwards compatible to v5, but not to earlier versions.

- Active4D cannot be used out of the box as a component because limitations in the plugin API prevent Active4D from accessing variables and methods in components.

Changes

v6.3r2

Changed components:
- Plugin
- Documentation
- Shell
- Demo

- A new configuration option, “parse json request” has been added. If this option is true, when a request with Content-Type “application/json” is received, the body of the request is parsed as JSON and the result is put in _form{“_json”}.

v6.3r1

Changed components:
- Plugin
- Documentation
- 4D shell
- NTK shell

- Now runs in 64-bit mode on OS X.

- Now runs under 4D v15r2 on Windows.

- [1179] URL decoding is more lenient and more compliant with the spec. If a ‘%’ is not followed by two hex digits, no attempt is made to decode.

- A new plugin command has been added: **A4D Abandon session**. This command is the equivalent of the Active4D **abandon session** command, but can be applied to any session. The shell’s Session Monitor has been updated to allow abandoning a session.

- Fixed some issues in the NTK receive callback method.

Upgrading to v6.3:

- After installing Active4D v6.3, open the shell database corresponding to the TCP/IP layer you use (4D or NTK).
- Open the A4D_SessionMonitor form.
- Copy the Abandon button.
- Close the shell database.
- Open your database.
- Add a project method called “A4D_AbandonSession” with the following code:

```
C_LONGINT($err)
$err:=A4D Abandon session ($1)
```

- Open the A4D_SessionMonitor form.
- Paste the Abandon button into the form. It should appear in the correct place, to the left of the Done button.

If you are using NTK, do the following in addition:

- Open your database.
- Open the A4D_HTTPD_RequestHandler method.
- Select all text and delete it.
- Select “Import Method…” from the Method menu, and import A4D_HTTPD_RequestHandler.c4d from the Upgrade/v6.3r1 folder in the Active4D distribution.
- Close that method and do the same for the A4D_NTK_ReceiveCallback method,

importing the A4D_NTK_ReceiveCallback.c4d method.
- Restart your database.

v6.2r1

Changed components:
- Plugin
- Documentation
- Shell
- Demo

- Added support for custom session handlers, with a predefined session handler that provides database-based session storage. See “Installing the Predefined Session Handler” in the “Installation” chapter for more information.

- Added **A4D LOG MESSAGE** command, which is used in the predefined session handler.

- The shell and demo are now 4D v13 databases. Earlier versions of 4D are too unstable on the latest versions of Mac OS X.

v6.1r15

Changed components:
- Plugin
- Documentation
- Shell

- [1176] Cookie key parsing is much more lenient now. Invalid keys are cleaned up and transformed into a compliant form. See the “Request Cookies” documentation for more info.

- [1178] 4D v13.6 broke WEB SEND RAW DATA (again!). A4D_Init was modified to turn raw data off for that version.

v6.1r14

Changed components:
- Plugin

- [1173] Creating process/interprocess variables on the fly in a compiled database now works without using a compiler declaration.

v6.1r13

Changed components:
- Plugin

- Recompiled with Visual Studio 2013.

- Replaced deprecated calls on Windows.

v6.1r12

Changed components:
- Plugin

- Recompiled with Xcode 6 and latest libs.

v6.1r11

Changed components:
- Plugin
- Shell (Cors.ini, sed.a4d)
- Demo (Cors.ini, sed.a4d)

- Detection of session ids now works as documented. If there is no session cookie or session cookies are not being used, query params are checked for the session id. If that fails, form variables are checked.

- Clarified the documentation of Cors.ini to make it clear that URL matching is exact, not partial.

- sed.a4d now embeds the session id in the form, so it will work even if session cookies are not being used.

v6.1r10

Changed components:
- Plugin

- Fixed output of **hide session field** command.

v6.1r9

Changed components:
- Plugin

- Fixed **session to blob** command.

v6.1r8

Changed components:
- Plugin

- Scientific notation can now be used for numbers.

- JSON numbers are now correctly output and parsed with period as the decimal separator, regardless of the system locale.

v6.1r7

Changed components:
- Plugin
- Shell
- Documentation

- Removed thread locks in logging to avoid a deadlock on exit.

- Documentation for the Default Directory now correctly indicates that the path may fall back to the user’s directory if the system directory is not writeable.

- Removed “Connection: close” code from A4D_OnWebConnection and A4D_HTTPD_RequestHandler, it was basically wrong.

v6.1r6

Changed components:
- Plugin
- Standard libraries (a4d.console, a4d.debug)
- Documentation

- Previously, things would break on Windows if the user directory or database directory had non-ASCII characters in the path. This has been fixed.

- Added support for BLOB and Time arrays when running in 4D v14 or later.

- Added support for a BLOB text format in all library methods that might dump a BLOB.

- Added some 4D v14 constants, and removed constants that apply only to the UI and don’t make sense within the context of Active4D.

- float and long64 fields are not supported by the plugin SDK, so attempting to access fields of those types will now generate an error instead of silently doing nothing.

v6.1r5-1

Changed components:
- Shell

- WEB SEND RAW DATA broke (again) in 4D v14.2, the shell was modified to disable using that command under v14.2.

v6.1r5

Changed components:
- Plugin

- Fixed a condition where resource files were not released when the plugin was unloaded but 4D did not quit.

v6.1r4

Changed components:
- Plugin

- [1164] **SET QUERY DESTINATION** now works correctly with 4D >= v13.3.

v6.1r3

Changed components:
- Plugin

- There was a bug in the underlying Unicode library which caused the **trim** and **right trim** commands to fail to trim whitespace on the right. This has been fixed.

v6.1r2

Changed components:
- Plugin
- Documentation
- a4d.json library

- When using the extended syntax for **QUERY/QUERY SELECTION**, you may now use any *unary* expression for the conjunction (e.g. a variable or method call), and *any* expression that yields a string containing a valid value comparator (#, =, >, >=, <, <=, %) as the comparator. In addition, in multi-line queries, you may omit the conjunction on lines after the first and it defaults to &.

- **a4d.json.parse** called **parse json** with the wrong parameter order. This has been fixed.

- **add to json** with a text array that contained empty elements would output nothing (only a comma) for the empty elemets. “” is now correctly output.

v6.1r1

Changed components:
- Plugin
- Documentation
- a4d.json library

- The entire a4d.json library has been converted to a new suite of native JSON commands. Converting data to JSON is now 24x to 40x faster than the a4d.json methods, and JSON parsing is now 50x faster! The a4d.json library has now been reimplemented as a thin wrapper around the native commands, so it gains all of their speed. Existing a4d.json code will work as it did before, but you are encouraged to convert to native commands, as the a4d.json library will be deprecated in the future.

- The new **add selection to json** command (and **a4d.json.addSelection**) adds the ability to return the result of a row expression verbatim, without JSON encoding. This allows the use of nested json generators. See the documentation for **add selection to json** for an example.

- The new **parse json** command (and **a4d.json.parse**) now takes a regular expression that can be used to match JSON keys that might be dates, and that you would like to convert to a 4D date. See the documentation for more info.

- The **add response cookie** command now takes Secure and HttpOnly parameters to set the corresponding attributes in the cookie. See http://en.wikipedia.org/wiki/Http_cookies#Secure_and_HttpOnly for information on the Secure and HttpOnly attributes.

- Added **set/get response cookie secure** and **set/get response cookie http only** commands.

- Session cookies always have the HttpOnly attribute set for greater security.

- An empty expires value can be passed to **set response cookie** to ignore that value.

- A new **session cookie secure** option has been added. When true, session cookies are marked secure and will not be sent to the browser if the request is insecure. You should only set this option to true if your site requires https for all requests.

- [1148] You may now pass * as an option to **CREATE FOLDER** to recursively create directories, as 4D v13 does.

- [1158] Eliminated spurious error messages when running under 4D Remote and the default directory is a symlink/alias/shortcut.

v6.0r13

Changed components:
- Plugin

- [1151] Fixed DISTINCT VALUES failing with process arrays.

- [1154] Fixed a crasher when trying to retrieve session values.

Please note: QUERY WITH ARRAY with integer fields is broken in 4D, there is nothing I can do about it.

v6.0r12

Changed components:
- Plugin

- Fixed a condition where crashes could occur when quitting 4D.

- [1150] Fixed DISTINCT VALUES and QUERY WITH ARRAY failing with integer fields.

- Fixed the case where the same positional format placeholder with no type (e.g.

“{0}:{0}” would generate garbage output.

v6.0r11

Changed components:
- Plugin

- Fixed a crasher that occurred with a Volume Runtime client, where no 4D licenses folder was available.

v6.0r10

Changed components:
- Plugin
- Shell (A4D_Init method)
- Documentation

- **IMPORTANT!** On Mac OS X, the minimum version is now 10.6 (Snow Leopard).

- **IMPORTANT!** The Unicode library used by Active4D, ICU, has been updated. Please note that the ICU data now lives in Active4D.bundle/Contents/Resources, and has different names. If you were storing ICU data in a shared location, be sure to copy the new ICU data files.

- The %% formatting operator now supports named placeholders in conjunction with collections. In addition, default formatting for values now conforms to 4D’s default formatting for the system locale.

- Added the **upload to blob** command, which returns an upload in a blob. See the documentation for more information.

- The **round** and **trunc** commands now call directly into 4D, so the results are guaranteed to be consistent with 4D.

- In 4D 13.2, the 4D shell will use **WEB SEND RAW DATA**, as it was fixed in 13.2.

- Named constants were updated to incorporate additions and changes in 4D 13.2.

v6.0r9

Changed components:
- Plugin
- Documentation

- [1143] When using the %% format operator, the current system locale is correctly used.

- Miscellaneous documentation tweaks.

v6.0r8

Changed components:
- Plugin
- Shell
- a4d.debug library
- Documentation

- **ORDER BY FORMULA** now has an option to be executed within Active4D's context, which allows access to Active4D's full execution environment within the order by expressions. Please see the documentation for more info.

- **a4d.debug.dump selection** now has an option to display the record number for a table by using "#" as the field name.

- [1142] To prevent high CPU usage on Server when a Remote is using the console, the refresh delay has been increased to 3 seconds by default. The delay can be configured with the ConsoleRefreshDelay item in the A4D_Config list. To upgrade, follow these steps:

    1. 1 Open the A4D_Config list in the Tool Box.
    2. 2 Add an item "ConsoleRefreshDelay=3" (without quotes) to the list. Change "3" to the number of seconds of delay you want between console refreshes.

3. 3 Replace line 24 of the A4D_Console form method with:

```
SET TIMER(Num(A4D_GetParam
("A4D_Config";"ConsoleRefreshDelay";"3"))*60)
```

Please see “Tuning Console Refresh” in the documentation for more info.

v6.0r7

Changed components:
- Plugin
- **a4d.utils** library
- Documentation
- Sublime Text 2 package

- Added support for the **command name** command.

- Added new methods, **a4d.utils.selectionToCollection** and **a4d.utils.selectionRangeToCollection**. Please see the documentation for details.

- Clarified the documentation for **a4d.json.parse** to indicate that you should always use **::=** to assign the result.

- A customized Active4D package for Sublime Text 2 is now available. For details, see http://active4d-nabble.aparajitaworld.com/Sublime-Text-2-support-tp4655369p4655375.html.

v6.0r6

Changed components:
- Plugin
- Documentation

- If a duplicate library name is encountered during startup, an error will be logged and the server will not crash.

- The source file and line number of library parse errors is now correctly logged if a library is loaded during startup or during a refresh.

- If a library load error occurs during startup, the error is logged and startup continues.

- References to v5 in the documentation were updated to v6.

v6.0r5

Changed components:
- Plugin
- **a4d.console** library
- Documentation

- Fixed improper formatting of header in **a4d.console.dump session**.

- Fixed an internal bug that would occur on shutdown when iterating through an empty array.

- Updated the documentation for **set session array** to indicate that directly declaring an array in a session will not actually create the session.

v6.0r4

Changed components:
- Plugin
- Documentation

- Added support for a format when using <%=.

- Fixed some typos in the documentation.

v6.0r3

Changed components:
- Plugin
- Documentation

- Added direct support for the 4D commands **Average**, **Min**, **Max**, and **Sum**.

v6.0r2

Changed components:
- Plugin
- a4d.json library
- Documentation

- Active4D will no longer attempt to create the com.aparajita directory in the shared 4D folder at startup. This should eliminate startup crashes and deactivation of the plugin due to inadequate permissions on Mac OS X.

- Key files may now be placed in the user 4D folder, which is the folder returned by **Get 4D folder**(**Active 4D Folder**) in 4D. This allows you to put the key file in a user-accessible folder on systems where you do not have write access to the shared 4D folder. The user 4D folder is searched before the shared 4D folder.

- Added **a4d.json.parse** and **a4d.json.convertJSONDates** for parsing JSON into an Active4D collection.

- [1134] Fixed a crash that would occur in some circumstances if a stray percent operator was encountered.

- [1136] Fixed a crash that would occur if a process/interprocess pointer was created in a compiled database, or if a nil process/interprocess pointer was passed to a 4D method.

v6.0r1

Please see the main documentation for more information on each change.

Changed components:
- Plugin
- Libraries
- Documentation
- 4D shell

- Demo

- **IMPORTANT:** In versions through v5, collection keys were kept in alphabetical order, although officially no order was guaranteed. In v6 the keys are kept in no predictable order, so if you have code that relies on alphabetical keys, you will have to get the collection keys, sort them, and then use the sorted keys to access the collection items.

- **IMPORTANT:** The default platform charset on non-Japanese/Chinese systems is now “utf-8” instead of “mac” or “win”. If your source files are not in UTF-8 (which they should be!) and you did not set a platform charset in Active4D.ini, you will have to set the platform charset now.

- **IMPORTANT:** The default output charset on non-Japanese/Chinese systems is now “utf-8” instead of “iso-8859-1”. If your output is not in UTF-8 (which it should be!) and you did not set an output charset in Active4D.ini, you will have to set the output charset now.

- The **session** command will create a session if it is used alone without referencing an item.

- The “X-Error-Status” and “X-Error-URL” headers are now put in the response headers as documented, instead of the former behavior of putting them in the request headers.

- Added the **get response status** command.

- Added **set http error page** and **get http error page** commands. Note that these act on a global basis, not per-request.

- The response status is maintained if you have custom execution/http error handlers.

- Within an error handler, the response status is initially HTTP Status OK (200). You can retrieve the error status that triggered the handler with the **get error status** command.

- **fusebox.handleError** (which used a redirect) has been deprecated in favor of **fusebox.handleErrorInline**, which does not redirect, thus allowing you to return a custom fusebox-based error page while maintaining the original context and response status.

- **throw** can now take a longint as the first parameter if you want to pass an error code. If you want to pass an error code and a message, the message can follow the code. The error message and error code passed to **throw** are now accessible via the **get throw message** and **get throw code** commands. The error code is also available in the $a4d_err_code variable or in the "a4d_err_code" item in $attributes if you use **fusebox.handleErrorInline**. Using codes makes it much easier to test for user errors in custom error handlers.

- fusebox now includes error codes when it throws errors. The various errors are defined as library constants.

- Within an error handler, the "fusebox page" option is ignored so that you are not limited to executing through fusebox.

- The demo now includes examples of the new error handling behavior.

- Previously, uploads of raw data were not detected properly. Now a POST is considered raw data if the Content-Type begins with neither application/x-www-form-urlencoded nor multipart/form-data.

- Most library methods that directly or indirectly dump a collection (such as **a4d.debug.dump collection**) can now take a filter argument for including or excluding items from the dump. The modified methods are:

  a4d.debug: dump collection, dump session, dump request info, dump query params, dump form variables, dump request, dump locals

  a4d.console: dump collection, dump session, dump request info, dump query params, dump form variables

- **a4d.debug.dump locals** now allows you to specify a named scope to dump, and allows filtering of the output.

- The shell methods **A4D_Init** and **A4D_OnWebConnection** were modified to work around a bug in **WEB SEND RAW DATA** in v13.

- In 4D v13, be sure to turn the preference "Launch Web Server at Startup" **off**.

The Active4D shell method **A4D_Init** will turn the web server on if necessary. If you leave the preference on, the web server will not turn on. This is actually a bug in v13, but the workaround is simply to turn the preference off.

- The session monitor has been redesigned using 4D listboxes instead of grouped arrays.
- If you pass the name “-” to **a4d.debug.dump array** or **a4d.debug.dump collection**, it will suppress the name header in the dump.
- **is a collection** and **is an iterator** will no longer generate an error if the parameter is not a number. This eliminates the need to test the type of the parameter first.
- **set local** has an added option that allows you to copy an array to the local instead of assigning the array’s current value.
- A new operator, **::=** (known as “super assign”), has been added. Super assign is like the normal **:=** operator but can only be used to assign to variables or collection items, and completely replaces the original variable’s value with a copy of the value to the right of the operator. This allows you to copy arrays by assigning to a variable directly!
- The shell and demo have been recreated as native 4D v11 databases, instead of being converted 2004 databases with legacy compatibility options.
- In Unicode mode, the console can now hold up to 2GB of text instead of 32K.
- Added the method **a4d.console.clear**, which clears the console.
- Added a read-only global collection accessible through the command **configuration** which contains the complete set of configuration information read from Active4D.ini, VirtualHosts.ini, Realms.ini, and ExtensionMap.ini.
- Added the method **a4d.debug.dump configuration**, which dumps the **configuration** collection.
- Added some extensions to ExtensionMap.ini and fixed the mime type for JavaScript files to application/javascript.

- By default, **a4d.debug.dump array** will hide the zero element. An extra parameter has been added to show it if desired.

- Added a “doctype” option in Active4D.ini that indicates whether pages are html or xhtml. This value is used in various places to build valid markup.

- Added a “*doctype” item to **request info** that gets its value from the “doctype” option.

- Added **a4d.web.br** method, which returns “<br>” for html doctypes and “<br/>” for xhtml doctypes.

- Added **a4d.web.emptyTag** method, which closes the tag with “ />” for xhtml doctypes.

- All library methods that use empty tags use **a4d.web.br** or **a4d.web.emptyTag** so valid markup is generated.

- Switched to HTML from XHTML in error pages and the demo, it turns out XHTML is pretty useless.

- Added **fusebox.invalidAction** method, a convenience method to call in the **else** of a switch. It throws an appropriate error describing what happened.

- [1122] When the request URL ends in a dot, Active4D now returns the custom http error page instead of the default 404 error page.

- **extension of("foo.")** now correctly returns an empty string instead of “.”.

- [1129] If you added a picture or picture array to a global collection and did not clear the collection, 4D would crash on quit. This has been fixed.

- [1027] Interchange with ObjectTools objects has been implemented via three new commands: **collection to object**, **object to collection**, and **clear object**. You must be using ObjectTools v4.0r1 or later for this to work. See the documentation for more info.

- Removed all vestiges of “4dcgi” from the demo, it was no longer necessary beginning with v11.

- Added support for more response statuses: 201 Created, 206 Partial Content, 405 Method Not Allowed, 501 Not Implemented, and 503 Service Unavailable.

- A response code of 204 No Contents correctly returns a Content-Length of zero and no response body.

- Added full support for CORS (Cross-Origin Resource Sharing) in the server and configuration via a new configuration file, Cors.ini.

- A POST with no body will no longer generate a 400 Bad Request error.

- Added an "allowed methods" configuration option to set which http methods are supported.

- Added support for the PUT and DELETE http methods, so Active4D can now be a full-fledged RESTful server.

- Within 4D, you may now open the Active4D console simply by calling **A4D_Console**.

- By default non-executable files served by Active4D will be returned with an expires date of 1 year in the future to allow the browser to cache it. The expires time can be set with the new **nonexecutable expires** option in Active4D.ini.

- The **expires** option is now set and logged properly.

- The filter box of the console prefs now allows you to enter IP addresses which contain zero as a component of the address.

- Fixed a bug that prevented Active4D from starting up under 4D Remote when the user logged in using a network account.

- The **DELETE FOLDER** command will now recursively delete folder contents if the * option is passed.

v5.0r37

Changed components:

- Plugin

- Fixed a crasher that could occur if the web root was outside the 4D default directory.

v5.0r36

Changed components:
- Plugin
- Documentation

- [1119, 1123] Fixed some documentation typos.

- [1122] Queries on time fields were totally broken. This has been fixed.

- This version will support the 64-bit version.

v5.0r35

Changed components:
- Plugin
- a4d.json library
- Documentation

- Fixed a regression that caused session cookie paths and domains to be incorrect.

- Added a new method, **a4d.json.writep**, which facilitates using the JSONP protocol.

v5.0r34

Changed components:
- Plugin

- A condition that would allow an attacker to hang the server has been fixed.

v5.0r33

Changed components:
- Plugin

- [1118] Fixed a regression in setting individual response cookie attributes.

v5.0r32

Changed components:
- Plugin
- ExtensionMap.ini

- [1117] Fixed new sessions replacing abandoned sessions getting lost.

- [1109] Added .7z to ExtensionMap.ini.

v5.0r31

Changed components:
- Plugin

- [1108] The boolean value False was converting to a string as the localized version of “True” instead of “False”. This has been fixed.

v5.0r30

Changed components:
- Plugin

- [1106] **param text** patterns which contained “$” followed by a digit would drop those characters. This has been fixed.

v5.0r29

Changed components:
- Plugin

- r28 broke things and was withdrawn, this release fixes those problems.

v5.0r28

Changed components:
- Plugin
- Documentation

- [1101] **find in field** no longer crashes in a compiled database.
- [1102] **base64 decode** no longer crashes if non-ASCII characters are passed to it.
- [1103] **get request infos**($names; $values) now works.
- **Substring** and **Delete string** now allow you to pass a negative value for the *inHowMany* parameter, which signifies an index relative to the end of the string. Please see the documentation for more info.

v5.0r27

Changed components:
- Plugin

- Fixed another crash in **replace string**.

v5.0r26

Changed components:
- Plugin

- Fixed a crash in **replace string**.

v5.0r25

Changed components:
- Plugin
- Documentation

- Text matching has been changed to work exactly the same way as 4D. Previously you could not search for CR or LF separately if they appeared together in the text. This has been fixed.

v5.0r24

Changed components:
- Plugin
- Documentation

- Pointer operations and commands with process/interprocess variables were very broken in a compiled database. This has been fixed.

- Changed the behavior of **Get pointer** and **RESOLVE POINTER** slightly to be consistent with Active4D variable naming conventions. Please see the documentation for more information.

v5.0r23

Changed components:
- Plugin
- Documentation

- Passing text with linefeeds or carriage returns to **base64 decode** will no longer crash. Now all whitespace is ignored.

- Updated the documentation to indicate the change to **base64 decode** and the fact that **continue** will increment a **for each** loop.

v5.0r22

Changed components:
- Plugin
- Documentation

- [1094] Fixed queries on boolean values in non-English versions of 4D.

- On Windows it is no longer necessary to install the Microsoft runtime libraries.

v5.0r21

Changed components:
- Plugin
- NTK Shell
- VirtualHosts.ini

- [1090] Moved a line of code in NTK shell to avoid a possible error.

- [1091] <number> * <text> now works.

- Fixed a problem with the NTK receive callback in compatibility mode.

- When in debug mode, the log file will display the configured virtual host name.

- **IMPORTANT CHANGE:** You may now use tabs or spaces between fields in VirtualHosts.ini. If any of the fields in your virtual host entries currently use spaces, they will have to be changed so they do not use spaces.

v5.0r19

Changed components:

- Plugin
- Documentation

- [1089] Hopefully fixed the default directory to be in the documented place under Windows.
- Fixed a bug where the Active4D library could not be found when running under Client.
- Fixed the documentation to correctly reference the section on the default directory wherever the default directory is mentioned.

v5.0r18

Changed components:
- Plugin
- Documentation

- [1086] Fixed an edge case where **replace string** would not correctly replace a series of two or more LF with CR.
- Fixed a bug in image.a4d that prevented images from being retrieved from the database if the id field was alpha.
- Documented the difference in string matching behavior between Active4D and 4D at the beginning of the “Strings” section of the Command Reference.
- Documented the need to declare all callback methods in the host database.

v5.0r17

Changed components:
- Plugin
- Documentation
- a4d.utils.a4l

- Converting a boolean value to a string (either automatically or via the **STRING** command) will respect the language version of 4D. For example, in the French version of 4D, **STRING(True)** returns “Vrai”.

- **QUERY/QUERY SELECTION** now works correctly with boolean values in non-English versions of 4D.

- The session id variable (*$sid* by default) is now properly created.

- [1082] Added missing **end case** in a4d.utils.truncateText.

- Small documentation tweaks.

v5.0r16

Changed components:
- Plugin

- [1080] Fixed problems with commands (such as **SELECTION TO ARRAY**) that required the creation of temporary process variables internally.

- The default web root under Client was set to the 4D application instead of the documented default directory. This has been fixed.

- **GET/SET DATABASE PARAMETER** was not working reliably. This has been fixed.

- Unicode/compatibility mode is now logged with the 4D environment.

v5.0r15

Changed components:
- Plugin

- [1079] Date subtraction was off by one in some cases, this has been fixed.

v5.0r14

Changed components:
- Plugin

- [1070] Log files are now rotated properly on Windows.

- [1072] ICU data is now found successfully on Windows and under Client.

- [1075] Boolean values (such as ‘auto create vars’) now work correctly in Active4D.ini.

- [1076] Using **continue** in the last iteration of a loop now works correctly.

- [1077] The Mac version works (again) on Mac OS X 10.4.

v5.0r13

Changed components:
- Plugin

- Key files located in the <shared 4D licenses>/com.aparajita/Active4D folder will be found as documented. Previously I was mistakenly looking in <shared 4D folder>/com.aparajita/Active4D.

- Active4D will startup correctly when running on Client. Previously it would fail to find the ICU resources and would be deactivated.

v5.0r12

Changed components:
- Plugin
- NTK shell (method A4D_HTTPD_ThreadPoolStart)
- Documentation

- On Windows XP (and perhaps Vista/7), the Microsoft VC++ runtime support libraries are required to be installed separately. This is not a change, I thought I could get away without this but it seems I cannot. If you are getting a -1 error because Active4D cannot be loaded when launching your database, you must install the runtime libraries. The runtime library installer can be downloaded here:

  www.microsoft.com/downloads/details.aspx

- [1068] **regex replace** with a subject array and replace array no longer crashes.

- The thread worker stack size was being set in bytes instead of kilobytes, which resulted in the minimum stack size, which was not enough. The stack size is now correctly set to the configured size in kilobytes.

v5.0r11

Changed components:
- Plugin

- [1063] 4D **Num** command now works correctly on non-US systems.

v5.0r10

Changed components:
- Plugin
- Documentation

- [1061] To improve compatibility, the v4.5 behavior of **QUERY BY FORMULA/ QUERY SELECTION BY FORMULA** has been restored and is now the default. To execute the command on the server within 4D's context, the first argument should be *. Please see the documentation for more details.

v5.0r9

Changed components:
- Plugin
- Documentation
- Active4D.ini

- [1057] Attempting to run or include an empty file would cause memory corruption on Windows. This has been fixed.

- [1060] Response cookies are now converted to UTF-8 and url encoded as documented.

- The documentation (in both Active4D.ini and in the reference) has been updated to indicate that you *may* prefix boot volume paths with "/Volumes/<volume name>/" on Mac OS X.

v5.0r8

Changed components:
- Plugin
- a4d.debug.a4l
- a4d.json.a4l
- a4d.utils.a4l
- Documentation

- Added support for **LONGINT TO BLOB** and **BLOB to longint** commands.

- Added the named constants **Intel byte ordering** and **PPC byte ordering** as aliases for **PC byte ordering** and **Macintosh byte ordering** respectively. The latter names are misleading, since all Macs now natively use "PC" byte ordering.

- **collection to blob** and **session to blob** have been enhanced to allow you to pass an existing blob, to which the serialized data is appended. Please see the documentation for more info.

- **blob to collection** and **blob to session** have been enhanced to allow you to pass an offset at which the serialized data is retrieved. Please see the documentation for more info.

- New library methods **a4d.utils.collectionToBlob**, **a4d.utils.sessionToBlob**, **a4d.utils.blobToCollection** and **a4d.utils.blobToSession** allow you to save and restore a collection or session with all embedded collections. Please see the documentation for more info.

- **a4d.debug.dump array** has a new parameter which controls whether numeric array elements which are collection handles are displayed as collections or not. Please see the documentation for more info.

- **a4d.debug.dump collection** has a new parameter which controls whether arrays are displayed inline or not. If they are not displayed inline, then numeric array elements that are collection handles are displayed as collections. Please see the documentation for more info.

- [1048] The *inCharset* parameter to **write blob** is now functional and the command does not crash when the type is binary.

v5.0r7

- [1043] Previously the http server was parsing cookie values according to the relevant specification (RFC 2965 and RFC 2616). But it turns out that in real life cookie values often break the rules (e.g. google analytics!). So now the server will accept anything as the cookie value.

  At the same time the cookie parser allows for cookies with a name but no value, as that is allowed in the spec.

v5.0r6

- [1040] Fixed **a4d.json.addSelection** and **a4d.debug.dump selection** to work correctly if there are deleted fields.

- [1041] Fixed documentation for **a4d.lists.findNoCase**.

- [1042] Character references now work with process variables.

- Updated the TextMate bundle: tweaked some snippets, added some new v11 commands to the syntax highlighting.

v5.0r5

- Fixed an assignment in **a4d.json.endArray**.

- [1033] All identifiers may begin with a digit to be consistent with 4D (although not with 4D’s documentation).

- [1034] Keyword queries (using the % comparator) are now supported.

- [1035] Time values may now be assigned to longint arrays.

- [1037] A boolean expression on the left side of the + operator, when the right side is text, will auto-convert the boolean expression to text.

- [1038] The server does a better job of matching RFC 3986 when parsing URLs, and if an illegal character in encountered it logs the character along with the error message.

v5.0r4

- Increased the minimum version of 4D to v11.5.

- [1028] Fixed a number of issues relating to process/interprocess arrays.

- [1032] **goto selected record** no longer restricts the index to >0.

- **regex replace** will accept the same variable/field for the subject and result.

- Changes to the web root are only logged when Active4D.ini is loaded or if the path is invalid.

v5.0r3

- [1028] **sort array** would crash under certain circumstances. This has been fixed.

- [1029] Passing an empty blob to **regex replace** no longer crashes, and the regex replace commands now check to make sure the subject and result are not the same variable/field.

v5.0r2

- [1026] **A4D Blowfish decrypt** now correctly removes padding.

v5.0

- [1019] A typo was fixed in **a4d.utils.deleteSelection**.

- [1020] Whitespace at the end of a path in Active4D.ini is now trimmed before the path's validity is checked.

- [1021] The output charset configured in Active4D.ini is now properly returned by the **get output charset** command.

- [1022] Assignment to different but type-compatible process variables works correctly in compiled mode.

- [1023] Crashes when using empty pointers have been eliminated.

- [1024] Using **append to array** with a variable for the value no longer kills the variable.

- The OEM variable linked to the key file is correctly recognized now.

- Removed some stray lists that were left in the NTK shell.

- Removed references to *A4D_ResponseHeaderNames* and

A4D_ResponseHeader_Values in compiler methods.

- Replaced some process arrays with local arrays in the NTK shell.

- **base64 encode** and **base64 decode** have been enhanced with a parameter that creates URL-safe encoding. Please see the documentation for more information.

v5.0MC2

- In compiled/compatibility mode, under circumstances text arrays would not be updated correctly. This has been fixed.

- Applied a fix to the 4D plugin SDK that was causing a memory leak.

- Added missing installation step in the documentation to setup the *On Web Connection* method.

- [1018] Table names starting with more than one digit are allowed.

v5.0MC1

- [949] **for each** now works correctly with process/interprocess variables.

- [977] Fixed a typo in **a4d.web.getEmptyFields**.

- [1007] Cleaned up some variables in the shell.

- [1008] Event handlers are not purged when using **A4D FLUSH LIBRARY**("@").

- Fixed a number of typos in the documentation.

- _The default image format for images loaded from the picture library and from methods by image.a4d is PNG instead of GIF.

- **_get license info** and **A4D GET LICENSE INFO** always return a Posix key file path.

v5.0b4

- [998] **delete collection item** now works with the **_request** collection.

- [1001] Modifying a library while 4D was running will no longer crash.

- [1004] **write to console** will now work with text that contains embedded tabs.

- [1006] Table names may now begin with a digit.

- **request info**{"*ajax"} is now a boolean. If the request is an Ajax request, it will be True, otherwise False, instead of "1" and "0".

- All known leaks in Active4D have been eliminated. As of v11.5 MC1, 4D leaks when using the following commands: **scan index**, **relate many selection**, and **picture properties** when used with a gif image read from disk. If you use these commands a lot, you should probably plan on restarting 4D once per day until these leaks are fixed.

  Previous versions of 4D leak a lot more.

- Adding to a date would be off by the time zone difference between GMT and the local time zone. This could result in a wrong day.

- The variable <>A4D_DialogTable is no longer used and has been removed from the shell, since dialogs don't need to belong to a table any more.

- **A4D_HTTPD_RequestHandler** in the NTK shell now properly checks for a valid table number when looping through the tables.

v5.0b3

- [991] Calls to **A4D FLUSH LIBRARY**("*") will no longer crash.

- You may now pass "@" to **A4D FLUSH LIBRARY** to flush all libraries except the Active4D library.

- Calls to **A4D FLUSH LIBRARY** are no longer synchronous. Please see the v5 Addendum for more information.

- Saving or loading a collection with an empty string item will no longer crash.

- The default web root is now <default>/web instead of <default>.

- All references to **ARRAY STRING** and **C_STRING** have been converted to **ARRAY TEXT** and **C_TEXT** in the shells and demo.

- General cleanup of the NTK shell, there was junk left there from testing.

- **type descriptor** now returns the type name for real arrays.

v5.0b2

- The **Date** command has been enhanced. An additional syntax is now:

  Date(year; month; day)

  This allows you to construct a date programmatically without playing tricks like constructing a date string or using the idiom:
  Add to date(!00/00/00!; year; month; day).

- The log files are now kept in <database structure directory>/Logs/Active4D. The “Logs” directory is what would be returned by **Get 4D folder**(Logs Folder).