

## Table of Contents

Edition Notes .....	3
Version 1.5.0 .....	3
Access Jump Start (AJS) architecture .....	3
Options Manager .....	3
Session Manager .....	4
Log Manager .....	4
Menu Manager.....	4
Error Processing .....	5
Source Code Versioning ( <i>optional</i> ) .....	5
AJS Library and Template Core Properties and Methods.....	6
Appendix A: Suggested Development Environment tools (Optional) .....	7
General Tools .....	7
MZ-Tools .....	7
vbWatchdog .....	7
Versioning Tools .....	7
Oasis-SVN.....	7
TortoiseSVN .....	8
SVN .....	8



*Access JumpStart*

AJS Overview (R 2.0.0)

## Edition Notes

### Version 1.5.0

1. Changed some default behaviors: turned off Menu form Resizing, turned off Debug.print startup messages, startup timing displays, getWindowsLogin() debug message.  
Set AJSAppSessionUtil.AJS\_ShowStartTime = True to display component startup timing. Function bootAJStimerElapsed() can be used by application code to display elapsed time and/or return the number of milliseconds since start of the application session.
2. New methods to Link and Unlink Back End databases. You can have an arbitrary number of Back End databases of varying types (MS Access or ODBC), each with an arbitrary number of linked tables. See the new "Back End DB Connections" dialog to manage the Database and Tables settings.
3. New option to write Session logs to a table in the PRIMARY Back End database.

## Access Jump Start (AJS) architecture

The infrastructure is a "skeleton" Access application with five independent subsystem "managers" plus a library of support modules to enhance VBA code development for Forms and Reports. It supports both 32-bit and 64-bit installations of Access.

The Framework consists of three Access files:

- Front-end (FE) and back-end (BE) database.  
It is built to support multi-user applications.  
All shared data resides in the back-end database. There is one Primary back end database defined and managed by AJS. You may also define and manage an arbitrary number of additional back end databases using the AJS.DbLink and AJS.DbUnlink methods.  
The front-end database contains all forms, reports, modules, macros, etc. and temporary tables specific to each user session. Each user executes their own copy of the front-end.
- The AJS Library database file.  
Most of the AJS infrastructure is contained in AJS\_Library.accdb. You may want to develop your application using the .accdb version of the file so that you can more easily test / debug. Then you can create a .accde version to distribute with your application if you like.

## Options Manager

Option parameters are defined to control the various operational behaviors of the target application. They are referenced as properties of the AJSopt class module.

Each option name can be read from a property of the same name. So, for example, you could print out the name of the current customer by entering:

```
debug.print AJSopt.AJS_CustomerName
```

- The developer specifies options in one place to control operation of the Framework VBA code.
- Developer-built VBA code can also refer to these parameters.
- In addition, you can add new application-specific Option Parameters as desired. See the AJS How To document for this and more.

The following infrastructure components can be configured for use in various combinations. They are controlled by the developer by referencing the methods and properties of the AJS class module.

## Session Manager

This component captures the user login information and sets up the session according to their Role. It logs the begin and end of the user session and records timestamps in tblAJSusers.

- The user login information is either captured from the present Windows Login or the user can be prompted for their stored User ID and Password.
- It can reject users not already in the database or allow them in with a default Role value.
- It can also check the location of the Front End and, if needed, copy it to a specified location for use only by the current user and then exit and restart using the new copy.
- If the Front End is already in the right place, it can check to see that it is the latest version and refresh it if not.
- You can use the User Administration menu to determine when and where each user logged in/out, who is currently logged in, and display/update which Roles can access each menu item.

## Log Manager

The optional Log Manager is used by Session Manager to log begin and end events of each session. Menu events are optionally logged by the Menu Manager. It is also used by Error Manager to log diagnostic information when an error trap occurs.

It presently can operate in 1 of 3 modes:

- Generate multiple windows text files (one per user session) to the defined shared directory, or
- Write all log entries to a table in the Primary Back End database, or
- Logging functionality can be disabled.

You can write a log entry from your VBA code by using the AJS.putLog method.

## Menu Manager

The AJS Menu feature is driven as part of the normal application start up process. First, an Autoexec macro calls function AJSstartup(). You can insert your customization VBA code in AJSappSessionUtil.AJSuserStartup() as noted in the comments. If you wish to modify the AJS Menu or bypass it completely, then just provide a menu form name in the AJS\_StartupMenu option and/or modify the code in AJSuserStartup() if more extensive logic is needed.

The menu system allows multiple menus each with any number of entries and provides a restricted view of menu items based upon user Role. If you make a copy of one of the standard menus to create your own custom menu.

Forms are provided to

- set user authorization and
- display the most recent login information for each user.

## Error Processing

The Error Processing module can provide global error processing for unhandled as well as handled error traps. It is designed to be used with or without the *optional* [vbWatchdog](#) third party product.

- When the *optional* [vbWatchdog](#) third party product is installed, this module will display and log error messages for all unhandled exceptions as well as handled ones. Each error trap is also displayed in the VBA immediate window when in Development mode.
- Without the [vbWatchdog](#) product installed, code must be inserted into each VBA Subroutine and Function in order to handle errors and call the Error Processing module. The *optional* [MZ-Tools](#) utility product can help with code insertion and line numbering in this case.

## Source Code Versioning (*optional*)

The Framework supports *optional* source code versioning with a [Sub Version](#) repository. It *optionally* uses [OASIS](#) and [TortoiseSVN](#) to interface between Access and the repository.

- Easily see the detailed differences between two versions of a module.
- See all the changes between two versions of an application or groups of selected modules.
- Create a record of each release of your software.

*This setup is not required, and you may manually enter a Version String for display instead.*

## AJS Library and Template Core Properties and Methods

You will use two modules to exercise the core capabilities of AJS:

1. **Module AJS**

is a Static Class Module that contains all necessary Methods and Properties to control the Rapid Deployment Framework. It can support AJS Error Manager, AJS Log Manager, AJS Session Manager and the AJS Menu System in any combination since it provides the developer interfaces for critical functions.

2. **Module AJSopt**

is a Static Class Module that exposes all the values of the Options table (tblAJSoptions) as properties, each of the same name.

When you type AJS or AJSopt you will be presented with a list of available properties and methods.

For example:

- AJS.PutLog is a method that writes a line to the active session log
- AJSopt.AJS\_CustomerName is a property that returns the Customer Name that is set in the Option table
- AJS.UserLogin and AJS.UserRole are properties that return the User Login and User Role for the current session respectively.

The items listed in document "HCI-AJS-Library-and-Reference" are intended for public consumption by the developer as well as by the AJS infrastructure. Other modules and functions whose names begin with "RDI" are part of the AJS Infrastructure and are not intended to be modified or invoked by the developer directly.

## Appendix A: Suggested Development Environment tools (Optional)

### General Tools

These tools are used at Halder Consulting, Inc. **We highly recommend them, but they are not required in order to use AJS.**

#### MZ-Tools

Available from <https://www.mztools.com/>, this product enhances the VBA development environment. Check their website for terms and pricing. Features include enhanced searching, formatting, sorting, code assistants (help entering common code sequences), private clipboards, Smart Design Mode, a code library and more. It is especially helpful for inserting often repeated code segments such as for error processing and for line numbering VBA code for enhanced error reporting if you do not use the [vbWatchdog](#) product.

#### vbWatchdog

Available from EverythingAccess.com, this product greatly enhances Access development debugging and is a platform for global error handling with minimal coding. See <https://www.everythingaccess.com/vbwatchdog.asp> for more information and to trial and purchase the product. You may use the trial version with AJS.

### Versioning Tools

Optional Source Code versioning is incorporated into the Framework by means of the following tools.

**Use of these tools is not required and is completely optional when using AJS.**

There are two major benefits to using Source Code version when developing Access applications:

1. When multiple developers are working on the same project, a good versioning system makes it much easier to partition the development and stay out of each other's way. If multiple users do modify the same module at the same time, the merge process is automated so that nobody's work is lost. In a worst-case scenario, the system will note and mark conflicting updates to make it as easy as possible to do a side by side comparison and manual resolution.
2. Even if just a single developer is working on a project, the versioning system makes it easier to "Tag" specific versions of the project to keep track of all changes. It reduces the number of backups that need to be made and makes it easy to recall a specific version of an application from the past.

#### Oasis-SVN

The OASIS\_SVN product provides seamless integration into the Access development environment, including the VB Editor. See <https://dev2dev.de/index.php?lang=en>. The product allows multiple developers to work on a single Access application concurrently. It supports automation, which HCI uses to update SVN version information. Check their website for terms and pricing.

According to their web site: Access "... Teamwork is only really possible by using a source code management system. With the help of OASIS-SVN, each team member can work on individual objects - with a version server on the Internet, even at any time at any location. Later, the individual objects from source code management are merged into a new version of the application."

[OASIS](#) also supports [GIT](#) and [Mercurial](#) back end repositories in addition to [SVN](#). We have not tested either of these at HCI. The Rapid Development Framework versioning automation supports only an SVN repository at present. (Please make your needs known and we will serve them to the best of our ability).

### TortoiseSVN

"Tortoise SVN is an Apache Subversion ([SVN](#)) client, implemented as a Windows shell extension. It's intuitive and easy to use, since it doesn't require the Subversion command line client to run. And it is free to use, even in a commercial environment..." from the web site: <https://tortoisesvn.net/>.

There are also Tortoise versions available for GIT ([TortoiseGIT](#)) and Mercurial ([TortoiseHg](#)).

### SVN

Apache Subversion ([SVN](#)) is typically hosted on a server. It is used for version control of various types of applications. See <https://subversion.apache.org/>. You can run the [SVN server on your PC](#), or there are hosted services available from third parties.

Note that for a single developer environment, you can use TortoiseSVN to create one or more local repositories on your PC and bypass the use of SVN completely. However, if it is likely that you will want or need to collaborate with other developers, you will need to host your work on an SVN server of some sort.