Adilas, LLC.
517 Blake Street
Salida, CO 81201
435.881.1536****

**Adilas.biz Concept Sign-Off Doc**

Use this page as a self-evaluation before going to do code sign-off:

* Does it work?
* Can you make it fail?
	+ Hit all if statements and error messages
	+ We param all incoming values
	+ We like to run validation on all incoming FORM and URL params
		- If using both FORM and URL, does your page switch or favor one scope or another? Is that clear where that switch happens?
	+ We Scope all variables – variables, FORM, URL, session, arguments, application, CGI, etc.
		- One exception is in CFC’s. We usually don’t use the variables scope (for local vars within a function). We do use the arguments scope in CFC’s.
	+ What about try/catch blocks and other general error handling?
		- Often we use try/catch blocks for queries, special logic, outside API socket calls, or places where we may not get everything we are looking for.
	+ How much data did you use for checking things? Is this typical of the live environment? What about pagination? Anything over 200 records needs pagination.
	+ Did you test for different values such as positives, negatives, zeros, strings, blanks, numerics, dates, and other possible hacks, etc.? Think different/random scenarios or different use cases.
* How are your comments?
	+ Did you add a modified page log at the top?
	+ We comment everything… this helps ourselves, other developers, and even our clients.
* Did you re-read it and clean it up?
	+ Check for extra dumps and aborts
	+ Whitespace
	+ Check for extra variables or un-used code
		- Did you copy and paste from another spot? Did you need everything that you copied?
	+ Alphabetizing things
		- We alphabetize params, method call arguments, queries (big selects, inserts, and updates), etc. If there are more than 3 values (fields, variables, arguments), we probably want them alphabetized.
	+ Trimming things
		- We trim in and out of all method calls
		- Any time we get URL or FORM scope stuff – basically any user input
		- All database inserts and updates. Keep the data clean and tight.
		- We don’t trim full structures or arrays. That causes problems.
	+ Naming convention
		- Do your variables make sense?
			* Recommendations: use all lower case, camelCaseStuff, or use\_under\_scores
			* All queries start with “qry” – qryGetCustomers or qryCheckTemplates
			* All database fields are all lower case with under scores
			* We like structures to have an “\_st” at the end – myTemp\_st or lineItems\_st
			* We like arrays to have an “\_array” at the end – my\_sample\_array or myMasterHolding\_array
			* We like lists to have an “\_list” at the end – temp\_id\_list or customerId\_list
			* Most CFC’s, methods, and function arguments have the camelCaseVersion of the actual field name (where possible). For example: the database field is first\_name, the argument would be firstName
			* Most page names are all lower case with underscores – index.cfm or add\_edit\_custom\_settings.cfm
			* Most action pages end in “\_action.cfm” – add\_edit\_time\_action.cfm or update\_settings\_action.cfm
			* Please watch out for reserved words or words that may be used in code. If needed, add the phrase “temp”, “current”, “my”, or “use” to help those values out. Example: useDate, tempStatus, currentId, etc. You get the idea.
	+ Did you retest it? Please make sure it still works after you have modified and cleaned it up.
* How tight and good are your queries?
	+ How are your joins?
	+ How many records are you expecting?
		- Did you check for 0 record count?
		- Did you check for a single record count?
		- Did you check for multiple records?
	+ Did you limit by corp?
	+ Did you limit by a status field?
	+ Remember single quotes for strings, CreateODBCDate stuff for dates, numerics are just plain, etc.
	+ Did you put an order by clause (if more than one possible record)?
* How is the formatting and CSS stuff?
	+ Does your code match the existing style and/or design of the system?
	+ All div tags and tables need classes. If the CSS class is unknown, use “basicText” as the class name. See this reference guide for available CSS classes:
		- <https://www.adilas.biz/css/>
	+ What about the different preset themes?
		- Does it work on the different themes?
			* Classic
			* Geometry, Blue Pastel, The Project, Armored Car
			* AFB Snow Owl – as a note, this one does go a little bit deeper for a more modern look and feel (mobile ready and such)
* On CFC’s (coldfusion components – methods and functions)
	+ Did you declare the local vars?
	+ All error messages are returned as a structure vs. a page include/abort.
	+ When calling a CFC or a method, use a path similar to: component="#application.pathToMainCFCs#.security\_2" – this is on normal .cfm pages and when components call other components.
	+ If changing a method, did you search to see where else the method is used?
		- Did you update all other occurrences of the method call?
		- What about in the Web/API land? Did you update things there?
* Did you script any database changes?
* What about documentation? Help files? Web/API documentation? Etc.
* Are your pages black boxed? If yes, did they follow the black box standards.
	+ If breaking pages into sub pages, you may need to add an additional ../ to any error messages in order to get the paths right. Please test and hit the error messages if breaking a page apart.
		- Example: <cfinclude template="../../message.cfm">
	+ Convert all CFC paths to the #application.pathToMainCFCs#.some\_page path (more details above in the normal to do list)
	+ Check for the word “black” to see if there are any older black boxes or code that needs some lovin’. Please fix and bring all older code into compliance.
	+ Check for “mainNavMenu”. If yes, change the div tag to:
		- <div id="mainNavMenu" style="display:none">
		- Get rid of the javascript at the bottom of the page that forces the display to hidden. This is for icon menus. The change to the div tag above will do the trick.
	+ When debugging the page, check to make sure that all eight standard black box pieces are being pulled in. See top\_secret/secure/GetSecurePaths.cfc for more info on black boxes. The normal eight are:

1. full - custom logic and/or a full take over - this is above any html tags
2. top - start of html to start of body tag - CSS, scripts, meta tag stuff
3. header - below the body tag - repeating logos and navigation for each page
4. top\_mini - optional - below the header and above the main page content
5. body - main meat and/or content for the page
6. bottom\_mini - optional - below the main meat or content and above the footer
7. footer - repeating logos and navigation
8. bottom - closing body and closing html tag – also any special scripts that are needed

* + Action pages usually only have two black box pieces. They are a full takeover and a bottom mini.
	+ If using the “getCustomLogic” method, please provide a sample of what the page path would look like. This is a custom black box option. See the top\_secret/secure/update\_cart.cfm page for samples. Just search for “getCustomLogic” to see a couple of usage examples.
	+ Also, make sure any date fields have the datepicker class applied. This helps us show the date pickers widget vs. static text fields. Sample:
		- <input type="text" name="start\_date" value="" class="datepicker" />
* Code Management - What about Git, Bit Bucket, Branches, and Pull Requests?
	+ Did we clean that up?
	+ Was it merged with master?
	+ Was it signed off?
		- At least two sets of eyes per page (peer reviewed)?
		- Tested before going live and hopefully tested on another machine?
		- Let’s make sure we are delivering good stuff.
	+ Was it pushed to live servers?
		- Which one(s)?
		- When was it pushed (we try to push in non-peak times – we have had disasters pushing right in rush hour traffic – especially for customer, invoice, quote, and cart stuff). Peak usage times are between 9 am and 9 pm. Best times are early in the morning or way after normal business hours. Some of the clients are open until midnight.