Identity Access and Management Web Site Design Overview

1. **Introduction**
   1. The Identity and Access Management (IDM) division is responsible for a vast array of employee maintenance actions in order to maintain an accurate access management environment.
   2. Many of these responsibilities rely on separate utilities working in unison to knit these actions together.
   3. The Identity Access and Management Web Site provides a ***One Stop Shopping*** platform to seamlessly tie these separate utilities together behind easy to use User Interface displays.
   4. The services provided within this site will be discussed in section II of this document.
   5. Access to the IDM Web Site can be achieved by clicking [here](http://idmgmtapp01.universal.co) or navigating to http://idmgmtapp01.universal.co
2. **Usage Overview**
   1. The majority of people who visit this site will do so with an intent that can be broken down into two categories:
      1. Informational
         1. Examples would be reviewing:
            1. HR contact information
            2. Master contact lists
            3. Departmental manning reports.
      2. Applications
         1. Examples would be executing:
            1. Terminating employees
            2. Adding new people into Active Directory
            3. Providing managers and other delegated personnel access to recently terminated Associate’s One-Drive accounts.
         2. All application are located within the **Admin Portal** portion of the website which is discussed below.
3. **Admin Portal**
   1. This is a secure area where only authorized who have access can perform applications listed in section II of this document.
   2. To gain access to the Admin Portal, the follow steps must take place:
      1. An invitation must be sent from the web site administration team to the individual requesting access.
      2. That individual will receive that invitation in the form of an e-mail.
      3. Once that individual clicks on the Register link located in the E-Mail, they will be forwarded to the registration area.
      4. Once registered, each user will be assigned an access level.
      5. These access levels control what applications each user has visibility into.
4. **Security**
   1. All associates who are granted access to the Admin Portal are passed an encrypted key to their computer which can only be deciphered by the Web Server.
   2. These encrypted key codes on their own are useless if obtained by an individual hacking into the laptop.
   3. Upon each login, a new encrypted key is regenerated and passed to the client’s laptop.
   4. If any suspicious behavior is observed, that individuals access to the web applications can immediately be suspended.
5. **Logging of Events**
   1. All actions performed within the Admin Portal are logged into a database table.
   2. Information within this table includes:
      1. Name and employee ID of individual who performed the work.
      2. Time the work are executed.
      3. Description of what work was performed.
   3. These logs can be easily reviewed at any time by the administrators of the web site.
6. **Job Flowcharts**
   1. Following the order of operations for a specific task such as adding a new user to the Admin Portal can be a comprehensive endeavor at best.
   2. This section discusses the flow of each task along with location details for each script as well as a description of it purpose.
   3. Here is a breakdown of each task:
      1. Accessing the Admin Portal

Step By Step logic path from beginning to end when Admin Portal button is clicked

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Starting Location [a] | Next Task [b] | Next Task [c] | Next Task [d] | Next Task [e] | Next Task [f] | Next Task [g] |
| Index.html | quicklinks.html | BuildMainSelectionButtons() | CreateMainHTMLResponse() | SendAdminPortalParameters() | CreateAdminPortalHTMLResponse.php | CreateAdminPortalHTMLResponse.pl |

Legend for Scripts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Letter | Script Name | Location | Database Used | Notes |
| a | Index.html | /htdocs/ | None | Main entry point into IDM webpage |
| b | quicklinks.html | /htdocs/webpages | None | Web Page that calls the button creation function |
| c | BuildMainSelectionButtons() | functions.js | None | Builds the buttons on the main page. |
| d | CreateMainHTMLResponse() | functions.js | None | Gathers cookie values to pass to SendAdminPortalParameters(). |
| e | SendAdminPortalParameters() | functions.js | None | Sends User and EncryptedKey cookie values to CreateAdminPortalHTMLResponse.php. |
| f | CreateAdminPortalHTMLResponse.php | /php/BuildWebpageScripts | None | Creates "IDM\_Applications/php/BuildWebpageScripts/AdminPortalMenuOutput.txt". |
| g | CreateAdminPortalHTMLResponse.pl | cgi-bin/IDM\_Applications/BuildWebpageScripts | None | Prints out the HTML code created by CreateAdminPortalHTMLResponse.php to screen. |

* + 1. Registering a new user from Administrator’s role

Step By Step logic path from the Administrator’s end when registering a new user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Starting Location [a] | Next Task [b] | Next Task [c] | Next Task [d] | Next Task [e] | Next Task [f] | Next Task [g] |
| HousekeepingTasks.html | HouseKeepingLinks.html | BuildHousekeepingSelectionButtons() | CreateHKHTMLResponse () | GetHKApplicationURL () | SendHousekeepingParameters () | CreateAddUserToPortalHTMLResponse.php |

Legend for Scripts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Letter | Script Name | Location | Database Used | Notes |
| a | HousekeepingTasks.html | /htdocs/ | None | Main entry point into Housekeeping Tasks |
| b | HouseKeepingLinks.html | /htdocs/ | None | Web Page that creates the Add User To Portal Button option |
| c | BuildHousekeepingSelectionButtons() | functions.js | None | Builds the "Add User To Portal" button among others. |
| d | CreateHKHTMLResponse () | functions.js | None | Checks to ensure user ISADMIN before invoking SendHousekeepingParameters(). |
| e | GetHKApplicationURL() | functions.js | None | Returns the Application URL used by the CreateHKHTMLResponse function. |
| f | SendHousekeepingParameters () | functions.js | None | Send illegalAccess field to CreateAddUserToPortalHTMLResponse.php. |
| g | CreateAddUserToPortalHTMLResponse.php | /php/BuildWebpageScripts | None | Creates the Add New User Input HTML page. |

* + 1. Accepting a registration from the end user’s role

Step By Step logic path from the end user when registering a new user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Starting Location [a] | Next Task [b] | Next Task [c] | Next Task [d] | Next Task [e] | Next Task [f] |
| CreateAddUserToPortalHTMLResponse.php | SendClientWebsiteInvite.pl | Register.html | RegisterNameToWebPage.html | CreateRegisterUserDropDown() | CreateRegisterHTML.pl |

Legend for Scripts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Letter | Script Name | Location | Database Used | Notes |
| a | CreateAddUserToPortalHTMLResponse.php | /php/BuildWebpageScripts | None | Creates the Add New User Input HTML page. |
| b | SendClientWebsiteInvite.pl | /IDM\_Applications | None | Sends the user an Invite E-Mail message to Register their name. |
| c | Register.html | /IDM\_Applications | None | Just the main \_Parent page from registration. |
| d | RegisterNameToWebPage.html | /IDM\_Applications | None | Appears when user clicks on Register link in the SendClientWebsiteInvite.pl script. |
| e | CreateRegisterUserDropDown() | functions.js | None | Launched by the RegisterNameToWebPage.html script, it creates the Drop-Down name list. |
| f | CreateRegisterHTML.pl | /IDM\_Applications | None | Prints out the IDM\_Applications\php\DisplayCreateRegisterHTML.php file to screen. |

* + 1. Launching one of the Admin Portal applications (ODD, Termination … etc)

Step By Step logic path from beginning to end when an Admin Portal application button is clicked

**Note:** Make this process like when the Admin Portal is clicked. This uses WebAdminPortalApplicationURL vice WebApplicationURL and replacesAdminPortalInitiate\_functions.js with AdminPortalApplications.js

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Starting Location [a] | Next Task [b] | Next Task [c] | Next Task [d] | Next Task [e] | Next Task [f] | Next Task [g] |
| topsidebar.html | BuildAdminPortalRefresh() | BuildAdminPortalSelectionButtons () | CreateHTMLResponse() | CreateGoodRequest() |  |  |

Legend for Scripts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Letter | Script Name | Location | Database Used | Notes |
| a | topsidebar.html | /htdocs/ | None | Main entry point into IDM webpage |
| b | BuildAdminPortalRefresh() | functions.js | None | Web Page that calls the button creation function |
| c | BuildAdminPortalSelectionButtons () | functions.js | None | Builds the buttons on the main page. |
| d | CreateHTMLResponse () | functions.js | None | Gathers cookie values to pass to SendAdminPortalParameters(). |
| e | CreateGoodRequest() | functions.js | None | Sends User and EncryptedKey cookie values to CreateAdminPortalHTMLResponse.php. |
| f | CreateAdminPortalHTMLResponse.php | /php/BuildWebpageScripts | None | Creates "IDM\_Applications/php/BuildWebpageScripts/AdminPortalMenuOutput.txt". |
| g | CreateAdminPortalHTMLResponse.pl | cgi-bin/IDM\_Applications/BuildWebpageScripts | None | Prints out the HTML code created by CreateAdminPortalHTMLResponse.php to screen. |



1. **Databases**
   1. Within the IDM website, there are thirteen databases that support the environment.
   2. These databases are:
      1. WebAdminPortalLoginDetails
      2. WebApplicationURL
      3. WebBuildHousekeepingButtons
      4. WebBuildMainSelectionButtons
      5. WebBuildSelectionButtons
      6. WebHousekeepingApplicationURL
      7. WebIDMWebSiteLoggedEvents
      8. WebLatestWebUserDTG
      9. WebMainApplicationURL
      10. WebNewUsers
      11. WebRegisteredUsers
      12. WebSearchFields
      13. WebStatusOfODDProgress
   3. Below is a description and breakdown of the field types for each database
      1. WebAdminPortalLoginDetails
         1. This tables registers each time a user (and it could be the same users coming in multiple times) logs on to the Admin portal.
         2. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| webUserDTG | Varchar | 15 | Timestamp for the users instance |
| userID | Varchar | 20 | User’s employee ID |
| lastName | Varchar | 50 | User’s last name |
| firstName | Varchar | 50 | User’s first name |
| IDActive | Bit | 1 | Identifies whether the users session is active for that timestamp |
| adminPortalLoginAttempt | Varchar | 15 | Successful or Unsuccessful based on whether they has access to the Admin Portal at the time |
| loginDTG | Datetime | N/A | DateTime stamp of when the user attempted to access the Admin Portal |

* + 1. WebApplicationURL
       1. Return the PHP HTML Output processing script for any given application.
       2. These HTML scripts are what the users sees based on conditions for things like application, access type .. etc.
       3. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| application | Varchar | 40 | Application name |
| applicationURL | Varchar | 255 | PHP path name |

* + 1. WebBuildHousekeepingButtons
       1. Supplies the fields to build out the link statement for each button in the main part of the Housekeeping page
       2. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| FunctionName | Varchar | 40 | Main function name to call |
| FunctionID | Varchar | 40 | ID of function I,e, OneDriveDelegation |
| MouseOver | Varchar | 40 | Function to call when cursor covers button |
| MouseLeave | Varchar | 40 | Function to call when cursor leaves button |
| Image | Varchar | 256 | URL of image |
| Width | Int | N/A | Width of button in pixels |
| Height | int | N/A | Height of button in pixels |

* + 1. WebBuildMainSelectionButtons
       1. Supplies the fields to build out the link statement for each button in the main part of the web page.
       2. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| FunctionName | Varchar | 40 | Main function name to call |
| FunctionID | Varchar | 40 | ID of function I,e, OneDriveDelegation |
| MouseOver | Varchar | 40 | Function to call when cursor covers button |
| MouseLeave | Varchar | 40 | Function to call when cursor leaves button |
| Image | Varchar | 256 | URL of image |
| Width | Int | N/A | Width of button in pixels |
| Height | int | N/A | Height of button in pixels |

* + 1. WebBuildSelectionButtons
       1. So, when I first started making links for the functions (these buttons were the first I created), I didn’t consider that there may be more function panels. I should have named this something like *WebBuildAdminPortalSelectionButtons*.
       2. These links are for the Admin Portal buttons (as you may have already guessed by my last comment).
       3. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| Function Name | Varchar | 40 | Main function name to call |
| FunctionID | Varchar | 40 | ID of function I,e, OneDriveDelegation |
| Value | Varchar | 40 |  |
| Image | Varchar | 255 | URL of image |
| Width | Int | N/A | Width of button in pixels |
| Height | Int | N/A | Height of button in pixels |

* + 1. WebHousekeepingApplicationURL
       1. Return the PHP HTML Output processing script for any given housekeeping application.
       2. Used for such things as adding new users, clearing SQL tables … etc.
       3. Below is the field breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| application | Varchar | 40 | Application name |
| applicationURL | Varchar | 255 | PHP path name |

* + 1. WebIDMWebSiteLoggedEvents
       1. This table is specifically for the One Drive Delegation application.
       2. It simply keeps track of the execution statistics such as when it began, percentage done .. etc.
       3. Below is the field breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| userID | Varchar | 20 | Terminated User |
| application | Varchar | 70 | OneDriveDelegation |
| Source\_user | Varchar | 70 | Terminated User |
| Target\_user | Varchar | 70 | Person requesting access to OD folder |
| OODAccessType | varchar | 10 | ADD or REMOVE |
| Time\_or\_execution | datetime | N/A | Time executed |

* + 1. WebLatestWebUserDTG
       1. This little guy links the application being processed with the user requesting the work.
       2. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| webUserDTG | Varchar | 20 | Terminated User |

* + 1. WebMainApplicationURL
       1. Return the PHP HTML Output processing script for any given application.
       2. This is used for the main screen, not the Admin Portal screen.
       3. These HTML scripts are what the users sees based on conditions for things like application, access type .. etc.
       4. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| application | Varchar | 40 | Application name |
| applicationURL | Varchar | 255 | PHP path name |

* + 1. WebNewUsers
       1. Used to add employees to the Admin Portal users list.
       2. These names will show up in the drop down list when going to the Register web page.
       3. Below is the field breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| EmpID | Varchar | 10 | Employee ID |
| Name | Varchar | 60 | Employees full name |
| AccessLevel | Int | N/A | Defines what apps they can see |
| Registered | Varchar | 3 | Yes or No. Set to yes when they register |
| Authorized | Varchar | 3 | Determines whether they have authority to use the Admin Portal |

* + 1. WewbRegisteredUsers
       1. This is used by the return text code authorization method.
       2. Currently we are using this method
       3. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| userID | Varchar | 20 | Employee ID |
| lastName | Varchar | 60 | Employees last name |
| firstName | Varchar | 60 | Employees first name |
| phoneNumber | Varchar | 20 | Employees phone number |
| Textcode | Varchar | 7 | Text code sent to user |
| userTextCode | Varchar | 7 | Text code entered by user |

* + 1. WebSearchFields
       1. This table is used specifically for the Terminate Associate application.
       2. It keeps track of the search string of employee number in case the user decides not to terminate the selected user and needs to go back to the previous search list of associates.
       3. Below is the field breakdown:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| webUserDTG | Varchar | 15 | Used to identify the user who is terminating an associate |
| srchAssocID | Varchar | 20 | Holds the employee ID search string |

* + 1. WebStatusOfODDProgress
       1. The table is used specifically for the One Drive Delegation application and is used to record progress on the screen.
       2. Below is the field breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Max Length | Description |
| pctDone | Varchar | 10 | Percentage completed |
| Msg | Varchar | 200 | Used to hold the current progress step description |
| Msg1 | Varchar | 100 | Holds the first of two completion messages |
| Msg2 | Varchar | 200 | Holds the second of two completion messages |