Asheville Master Script Upgrade

## Summary

Accela has upgraded the agency scripts in the Asheville agency from 1.x and 2.x formats to the newest scripting 3.0 format. The original scripts that were stored in standard choices have been migrated into JavaScript files, using the Accela best practice directory structure. All custom functions that were embedded in master script code have been analyzed and migrated into the custom function library. The standard choices have been converted to a structure that mirrors the record type structure for the agency. This document describes the details of the conversion.

## Document Control

| Date | Author | Version | Change Reference |
| --- | --- | --- | --- |
|  |  |  |  |
| 10/31/16 | John Schomp | 0.1 | Working Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Analysis

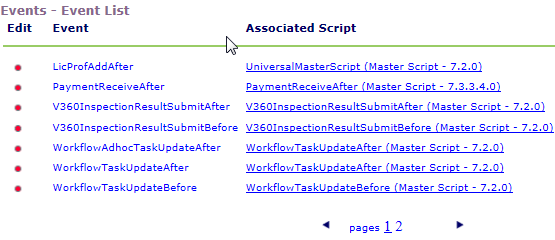
A review of the scripting shows that customary standards for scripting based on version 1.X have been used. “Variable branching” did not exist in these versions, so was not used.

Several opportunities for code review and/or refactoring were identified. Please see the Recommendations section for details.

## Master Script Usage

The following events are in use for the COSD agency, shown with associated master scripts.





## Edits to Standard Master Scripts

With minor exceptions, the master scripts in use are “Productized” which means that they cannot be edited. Two master scripts (FeeAssessAfter, LicProfLookupSubmitAfter) are non-productized versions from the V1.6 distribution. These have not been edited. The VoidFeeAfter script is not a master script – it performs hardcoded functions for this event.

## Custom functions that conflict with Master Script distribution

Custom functions were found that matched standard functions that are part of the Master Script 3.0 distribution. The code was compared and the following recommendations were made.

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Disposition of custom function** |
| addAdHocTask | Matches distribution | Removed |
| updateFee | Matches distribution | Removed |
| updateFeeItemInvoiceFlag | Matches distribution | Removed |

## 

## Functions with multiple versions

The following are custom functions that had substantial differences between versions across multiple master scripts. The recommendations are described below.

|  |  |  |
| --- | --- | --- |
| **Function\_Master Script Source** | **Description** | **Disposition** |
| addTimeAccountingRecord\_IRMA.js | Pulls date logged from inspection result, adds inspection details into notation. Fix to total minutes bug. | Combined into custom function |
| addTimeAccountingRecord\_IRSA.js | Does not contain IRMA, VIRSA updates | Combined into custom function |
| addTimeAccountingRecord\_VIRSA.js | Pulls date logged from inspection result, adds inspection details into notation. Fix to total minutes bug. | Combined into custom function |
| addTimeAccountingRecord\_WTUA.js | Custom workflow logic. | Combined into custom function |
| canScheduleDay\_IMSA.js | Fixes daylight savings bug, strict type checking on result | Using this version |
| canScheduleDay\_IRMA.js | no daylight savings fix, script type checking on result | Removed |
| canScheduleDay\_IRSA.js | no daylight savings fix, script type checking on result | Removed |
| canScheduleDay\_ISA.js | Fixes daylight savings bug, strict type checking on result | Removed |
| canScheduleDay\_VIRSA.js | Fixes daylight savings bug, does not do strict type checking on result | Removed |
| canScheduleDay\_WTUA.js | no daylight savings fix, script type checking on result | Removed |
| checkTAFee\_TAAA.js | Lacks checktype parameter. This version does not appear to be called from anywhere | Removed |
| checkTAFee.js | Has checktype parameter | Using this version |
| externalLP\_CA\_COSD\_ASIUA.js | Less error checking, no custom logic | Review |
| externalLP\_CA\_COSD\_LPAA.js | Has more robust error checking on template data. Uses CSLB URL Lookup. Custom record-specific logic 13ACC-06318. | Review |
| externalLP\_CA\_COSD\_LPUA.js | Less error checking, hardcoded URL. No custom logic. | Review |
| formatTimeAMPM\_old\_ADUA.js | Not referenced | Removed |
| formatTimeAMPM\_old\_ASIUA.js | Not referenced | Removed |
| getASBContacts\_ADUA.js | Custom DB logic needs to be reviewed by Agency | Review |
| getASBContacts\_ASIUA.js | Custom DB logic needs to be reviewed by Agency | Review |
| paymentByTrustAccount\_IRMA.js | returns true/false | Removed |
| paymentByTrustAccount\_IRSA.js | custom return codes, parseFloat on account balance and overdraft limit | Using this version |
| paymentByTrustAccount\_TAAA.js | custom return codes, no parseFloat on account balance and overdraft limit | Removed |
| paymentByTrustAccount\_TAUA.js | custom return codes, no parseFloat on account balance and overdraft limit | Removed |
| paymentByTrustAccount\_VIRSA.js | custom return codes, parseFloat on account balance and overdraft limit | Removed |
| paymentByTrustAccount\_WTUA.js | custom return codes, no parseFloat on account balance and overdraft limit | Removed |
| PDSBLDNextSchedDate\_IMSA.js | uses addDST function for daylight savings compatibility | Removed |
| PDSBLDNextSchedDate\_IRMA.js | Does not call addDST | Removed |
| PDSBLDNextSchedDate\_IRSA.js | Does not call addDST | Removed |
| PDSBLDNextSchedDate\_ISA.js | uses addDST function for daylight savings compatibility, better handling of parameter | Using this version |
| PDSBLDNextSchedDate\_VIRSA.js | uses addDST function for daylight savings compatibility | Removed |
| PDSBLDNextSchedDate\_WTUA.js | Does not call addDST | Removed |
| prescript140\_handler\_ASIUA.js | Handles 0 appending properly | Using this version |
| prescript140\_handler\_IRMA.js | Treats zero as integer which could cause issues | Removed |
| prescript140\_handler\_VIRSA.js | Handles 0 appending properly | Removed |
| script139\_HMR\_change2Active\_PPA.js | Comment describes "Payment Portlet", unsure of which is desired | Review |
| script139\_HMR\_change2Active\_PRA.js | Comment describes "Payment Tab", unsure of which is desired | Review |
| script139\_PRP\_change2Active\_PAA.js | Better error handling if "Registration Start Date" is null. | Using this version |
| script139\_PRP\_change2Active\_PPA.js | Does not handle null Registration Start Date | Removed |
| script139\_PRP\_change2Active\_PRA.js | Does not handle null Registration Start Date | Removed |
| taskCloseNonResulted\_ASUA.js | Less Comments | Removed |
| taskCloseNonResulted\_WTUA.js | Contains more comments | Using this version |

## Standard Choice Conversion Details

* Approximately 5,638 Standard Choices containing scripts have been converted to 201 script files. 99 new functions were created.
* In general, any standard choice code that was branched to more than once has been created as a separate custom function. In most cases the function name is the same or similar to the name of the original standard choice. For example:

if (inspType == 'Pre-Construction Inspection' && inspResult == 'Pass') {

//replaced branch(DPW\_ES\_LD\_PRECONST\_COMPLETE)

DPW\_ES\_LD\_PRECONST\_COMPLETE();

}

* All converted scripts have been tested for proper JavaScript syntax.
* To assist with testing, the original “Branch” standard choice names have been added as comments to the converted scripts. Once testing is complete these comments won’t be needed. For example:

if (checkCapForLicensedProfessionalType('CSLB Contractor')) {

//start replaced branch: ES\_CSLB\_AFTER

{

comment('Line 01 CLSB');

cslbMessage1 = null;

popUpMessageTbl = '';

* The conversion algorithm may have created some instances where double sets of braces {{ … }} denote a code block. These can be removed if desired, but they have no impact on the script execution.
* All references to showDebug within business scripts have been removed in order to simplify script debugging. Accela recommends setting this in one place, preferably using INCLUDES\_CUSTOM\_GLOBALS for this purpose. For example:

if (matches(currentUserID,'CHARSHBARGER','SJCHOMP’,’JOEC')) {

showDebug = true;

}

* Individual standard choices that were disabled are included but commented out inline. Any branch calls in this disabled code were migrated as “br\_nch” as the branch function is no longer supported in 3.0. If this code needs to be enabled, these branch calls would have to be refactored. For example:

copyParcelGisObjects();

// DISABLED: ApplicationSubmitAfter:9

//if (appMatch('LUEG-PDS/\*/\*/\*')) {

// br\_nch('ES\_PDS\_ApplicationSubmitAfter');

// }

// DISABLED: ApplicationSubmitAfter:16

//if (appMatch('LUEG-PDS/\*/\*/\*')) {

// br\_nch('ES\_PDS\_PROXIMITY\_ALERTS');

// }

* Standard choice branches that were not referenced by any script were converted into the Scripts\Events\Unused Branches folder. This includes any code that was referenced by disabled standard choices. These should be reviewed.
* All script files have been formatted to JavaScript standards for indentation.
* All references to disableTokens global variable have been removed. This flag is deprecated in scripting 3.0. Curly braces ({}) can (and should) be used to designate bode blocks in 3.0 scripts.
* The caret (^) is no longer valid for creating conditional logic. Standard JavaScript should be used instead (i.e., if, else, else if, switch)
* All references to ASI/TSI/Parcel Attributes using curly braces (e.g., {ASI Field}) have been changed to use the AInfo global Array (e.g., AInfo[‘ASI Field’]). Curly braces are now exclusively used to designate blocks of code.
* All references to the “branch” and “endbranch” functions have been removed. These are deprecated in Master Scripts 3.0.

## Code Analysis and Recommendations

Some recommendations on the existing code are as follows:

* There are some script lines used to create outgoing emails. There is new email template functionality that can be used which would make management of these emails much easier. See the V360 Admin -> Notification Template feature for more information. The number of scripts used to send these emails could be dramatically reduced.
* Review environment specific code and add references to global variables in INCLUDES\_CUSTOM\_GLOBALS that represent the environment (DEV, STAGE, PROD, etc.) so that code can remain consistent from env to env.
* Review/Edit all TODO sections as shown below:

## Script Repository

The master script 3.0 deployment for the County of San Diego is located in TBD

|  |  |
| --- | --- |
| Repository Site | [TBD](http://www.assembla.com) |
| Repository URL | TBD |

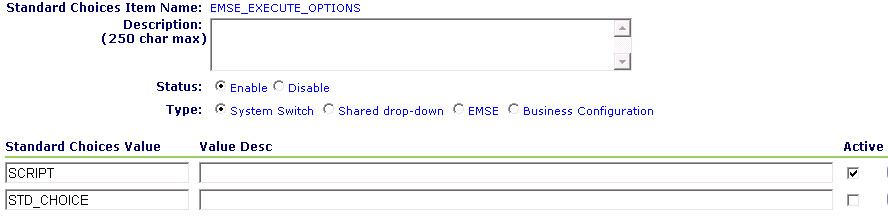
To connect to the repository, the “EMSEToolConfig” standard choice will need to be configured as per the Accela documentation.

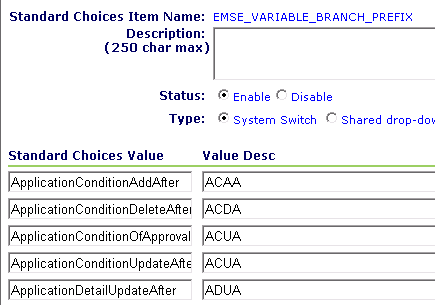
Once added, the “EMSETool” portlet can be added to an administrator console.

## Standard Choices

The following Standard Choices need to be configured. These are documented in the Master Script distribution in the documentation folder.

* EMSE\_EXECUTE\_OPTIONS determines if Scripts or Standard Choices, or both should be executed by the master script.
* EMSE\_VARIABLE\_BRANCH\_PREFIX determines the naming convention used to determine if a script is associated to a particular event.





Event Mapping required for COSD (highlighted are deviations from standard):

|  |  |
| --- | --- |
| ApplicationConditionAddAfter | ACAA |
| ApplicationConditionBatchUpdateAfter | ACBUA |
| ApplicationConditionUpdateAfter | ACUA |
| ApplicationDetailUpdateAfter | ADUA |
| ApplicationDetailUpdateBefore | ADUB |
| ApplicationSubmitAfter | ASA |
| ApplicationSubmitBefore | ASB |
| ApplicationSpecificInfoUpdateAfter | ASIUA |
| ApplicationSpecificInfoUpdateBefore | ASIUB |
| ApplicationStatusUpdateAfter | ASUA |
| ContactAddAfter | CAA |
| ContactEditAfter | CEA |
| ConvertToRealCAPAfter | CTRCA |
| FeeAssessAfter | FAA |
| InspectionResultModifyAfter | IRMA |
| InspectionResultModifyBefore | IRMB |
| InspectionResultSubmitAfter | IRSA |
| InspectionResultSubmitBefore | IRSB |
| InspectionScheduleAfter | ISA |
| InspectionScheduleBefore | ISB |
| LicProfAddAfter | LPAA |
| LicProfLookupSubmitBefore | LLSB |
| LicProfUpdateAfter | LPUA |
| ParcelAddAfter | PAA |
| PaymentReceiveAfter | PRA |
| PaymentReceiveBefore | PRB |
| TimeAccountingAddAfter | TAAA |
| TimeAccountingAddBefore | TAAB |
| TimeAccountingUpdateAfter | TAUA |
| TimeAccountingUpdateBefore | TAUB |
| WorkflowTaskUpdateAfter | WTUA |
| WorkflowTaskUpdateBefore | WTUB |

## Event to Script Mapping Prior to Upgrade

TBD

## Event to Script Mapping After Upgrade

TBD