Release Notes

Gray Quarter, Inc.

Gray Quarter

Master Script Upgrade Release Notes

Document Control

Date	Author	Version	Changes
01/02/2019	John Schomp	0.5	DRAFT

Leon County Master Script Upgrade

Summary

Gray Quarter, Inc. has upgraded the agency scripts extracted from the LEONCO Production environment from the version 1.x and 2.x format to the newest scripting version 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. These 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, as well as implementation steps.

Analysis

A review of the scripting shows that customary standards for scripting based on version 2.X have been used. Variable branching was used, and the script migration follows the branc

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 LEONCO Production agency, shown with associated master scripts. The "Action" column describes the changes that will be required when implementing the upgraded scripts.

Event	Script	Productized	Version	Action
ApplicationDetailNewBefore		N/A	N/A	Disable Event
ApplicationStatusUpdateAfter	ApplicationStatusUpdateAfter	No	2.0	Use Productized 3.0
ApplicationSubmitAfter	ApplicationSubmitAfterModified	No	2.0	Use Productized 3.0
ApplicationSubmitBefore	ApplicationSubmitBefore	No	2.0	Use Productized 3.0
CAEConditionAddAfter	CAEConditionAddAfter	No	2.0	Use Productized 3.0
ContactLookUpAfter		N/A		Disable Event
ConvertToRealCAPAfter	ConvertToRealCapAfter2.0	No	2.0	Use Productized 3.0
FeeEstimateAfter4ACA	FeeEstimateAfter4ACA	No	2.0	Use Productized 3.0
LicProfAddAfter		N/A		Disable Event
LicProfUpdateAfter	LICPROFUPDATEAFTER	No	2.0	Use Productized 3.0
PaymentReceiveAfter	PaymentReceiveAfterV2.0	No	2.0	Use Productized 3.0
RefLicProfAddAfter	RefLicProfAddAfter	No	2.0	Use Productized 3.0
RefLicProfUpdateAfter	RefLicProfUpdateAfter	No	2.0	Use Productized 3.0
VoidFeeAfter	VoidFeeAfter	No	2.0	Do not modify, not a master script
WorkflowTaskUpdateAfter	WorkflowTaskUpdateAfterV2.0	No	2.0	Use Productized 3.0
WorkflowTaskUpdateBefore	WorkflowTaskUpdateBeforeV2.0	No	2.0	Use Productized 3.0
ApplicationDetailUpdateAfter	UniversalMasterScript (Master Script - 7.3.0.0)	Yes	2.0	Use Productized 3.0
ApplicationSpecificInfoUpdateAfter	ApplicationSpecificInfoUpdateAfter (Master Script - 7.3.3.5.0)	Yes	2.0	Use Productized 3.0
ApplicationSpecificInfoUpdateBefore	ApplicationSpecificInfoUpdateBefore (Master Script - 7.3.3.5.0)	Yes	2.0	Use Productized 3.0
ApplicationStatusUpdateBefore	ApplicationStatusUpdateBefore (Master Script - 8.0.0.0.0)	Yes	2.0	Use Productized 3.0
ContactAddAfter	ContactAddAfter (Master Script - 7.2.0)	Yes	2.0	Use Productized 3.0
ContactEditAfter	ContactEditAfter (Master Script - 7.2.0)	Yes	2.0	Use Productized 3.0
InspectionMultipleScheduleAfter	InspectionMultipleScheduleAfter (Master Script - 7.3.1.2)	Yes	2.0	Use Productized 3.0
InspectionMultipleScheduleBefore	InspectionMultipleScheduleBefore (Master Script - 9.0.3)	Yes	3.0	Use Productized 3.0
InspectionResultSubmitAfter	InspectionResultSubmitAfter (Master Script - 7.3.3.5.0)	Yes	2.0	Use Productized 3.0
InspectionResultSubmitBefore	InspectionResultSubmitBefore (Master Script - 9.0.3)	Yes	3.0	Use Productized 3.0

InspectionScheduleAfter	InspectionScheduleAfter (Master Script - 7.2.0)	Yes	2.0	Use Productized 3.0
InspectionScheduleBefore	InspectionScheduleBefore (Master Script - 9.0.3)	Yes	3.0	Use Productized 3.0
LicProfLookUpSubmitBefore	LicProfLookupSubmitBefore (Master Script - 7.3.3.0)	Yes	2.0	Use Productized 3.0
V360InspectionResultSubmitAfter	V360InspectionResultSubmitAfter (Master Script - 7.2.0)	Yes	2.0	Use Productized 3.0
V360InspectionResultSubmitBefore	InspectionResultSubmitBefore (Master Script - 9.0.3)	Yes	3.0	Use Productized 3.0

Edits to Standard Master Scripts

- ApplicationStatusUpdateAfter was modified to work like a productized script (using productized include files)
- ApplicationSubmitAfter was modified to contain licensed professional interface code which has been extracted
- CAEConditionAddAfter was modified to contain LP condition code which has been extracted
- RefLicProfAddAfter was modified to contain LP code which has been extracted

Custom functions that conflict with Master Script distribution

Function	Action
addFee.js	Custom message, retain
addParameter.js	Remove
closeCap.js	Remove
convertContactAddressModelArr.js	Remove
convertDate.js	Remove
createRefContactsFromCapContactsAndLink.js	Remove
createRefLicProf.js	Remove
describe.js	Remove
describeObject.js	Remove
editRefParcelAttribute.js	Remove
generateReport.js	Custom, retain
getACARecordParam4Notification.js	Custom, retain
getACARecordURL.js	Custom, retain
getConditions.js	Remove
getContactByType.js	Remove
getContactParams4Notification.js	Remove
getParcelConditions.js	Remove
getPeople.js	Custom message, retain
getUserEmail.js	Remove
getWorkflowParams4Notification.js	Remove
handleError.js	Remove
include.js	Remove
inspCancelAll.js	Custom, retain
lookup.js	Remove
resultWorkflowTask.js	Custom logging, retain

sendNotification.js	Remove
updateFeeItemInvoiceFlag.js	Remove

Functions with multiple versions

- getFees_Inspections_Building.js has a difference in logic and needs review.
- sendLPExpiredNotification.js has a minor difference in email message, should be reviewed.

Standard Choice Conversion Details

- Approximately 2012 Standard Choices containing scripts have been converted to 92 script files. 17 new functions were created.

- 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 (wfTask == 'Quality Check' && wfStatus == 'Invoice Fees') {
    //start replaced branch: EMSE:AutoInvoiceBuildingFees
    {
        showMessage = true;
        feeItemArray = aa.fee.getFeeItems(capId, null, 'NEW').getOutput();
        if (feeItemArray) {
            for (FI in feeItemArray)
                //start replaced branch: EMSE:InvoiceTheFee
                {
                thisFeeCode = feeItemArray[FI].getFeeCod();
                thisFeeStatus = feeItemArray[FI].getFeeitemStatus();
                comment('Fee Code = ' + thisFeeCode);
```

- 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.

- In accordance with best practices, all references to *showDebug* within business scripts have been removed to simplify script debugging. Accela recommends setting this in one place, preferably using INCLUDES_CUSTOM_GLOBALS, which was designed for this purpose.

- 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:

```
// DISABLED: CTRCA:Building/*/*/*:0030
// if (cap.isCreatedByACA()  && appTypeArray[1] == 'VelocityHall') {
//  br_nch('EMSE:Add-Inspections-VelocityHall');
//  }

// DISABLED: CTRCA:Building/*/*/*:0040
// if (appTypeArray[1] == 'VelocityHall') {
//  br_nch('EMSE:Add-Inspections-VelocityHall');
//  }
```

- Standard choice branches that were not referenced by any script were converted into JavaScript and placed into the Scripts\Unused 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.

- 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 Custom Fields, Task Specific Info, and Parcel Attributes that used braces (e.g., *{ASI Field}*) have been changed to use the AInfo global Array (e.g., *AInfo['ASI Field']*). 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

- Review/Edit all TODO sections as shown below:

LEONCO\Scripts\Event\ASUA;AMS!STORM!DRAIN!CLEANING.js	Line 2: // TODO: JHS 1/2/19 Consider making a function as this code is run for several record types on this event.
LEONCO\Scripts\Event\IRSA;BUILDING!~!~!~.js	Line 1: // TODO: confirm sequence order
LEONCO\Scripts\Event\WTUA;BUILDING!~!~!~.js	Line 256: // TODO: there is a syntax error on this standard choice line 0410. Fixed for now.

Script Repository

A temporary repository was created for this conversion

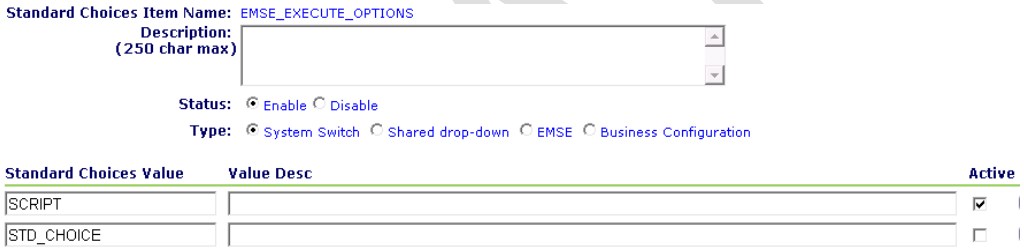
Repository Site	Github.com
SUPP Repository URL	https://github.com/grayquarter/LeonAccelaScripts.git/branches/SUPP

To connect to the repository, the "EMSEToolConfig" standard choice should be configured as per the Accela documentation. The "EMSETool" portlet should be enabled for the appropriate users/groups.

Standard Choices

The following Standard Choices must be configured. These are documented in the Master Script distribution in the documentation folder. The "Master Script Distribution_Std_Choices.zip" data manager package can be used to load these values.

- EMSE_EXECUTE_OPTIONS determines if Scripts or Standard Choices, or both should be executed by the master script. This should be set to select Scripts only.
- EMSE_VARIABLE_BRANCH_PREFIX determines the naming convention used to determine if a script is associated to a particular event.
- MASTER_SCRIPT_DEFAULT_VERSION (Recommendation is to set to 8.0.1.0.0)



Deployment to Higher Environments

After the initial migration, the deployment to a higher environment (e.g., DEV -> TEST, TEST-> PROD) can be achieved by the following steps:

	Step	Description	Est. Time
1	Create rollback package	In TARGET environment create a Data Manager export package that contains: • all Event to Script associations (AA Configuration, Event Manager, Events, select all) • Master Includes (AA Configuration, Event Manager, Master Scripts, INCLUDES_CUSTOM) • Includes (AA Configuration, Event Manager, Master Scripts, INCLUDES*) Save and run the job, saving the output in a dedicated folder. This package will only be used if a roll back is required.	5 Minutes
2	Create master script package	In SOURCE environment create a Data Manager export package that contains: • all Event to Script associations (AA Configuration, Event Manager, Events, select all) • Includes (AA Configuration, Event Manager, Master Scripts, INCLUDES*) • EMSE Standard Choices (AA Configuration, Standard Choices: EMSE_EXECUTE_OPTIONS, EMSE_VARIABLE_BRANCH_PREFIX, EMSEToolConfig) Save and run the job, saving the output in a dedicated folder.	5 Minutes
3	Deploy master script package	In TARGET environment create a Data Manager import job that will import the entire package in step 2.	5 Minutes
4	Confirm EMSEToolConfig	In TARGET environment update the EMSEToolConfig standard choice with credentials and URL to access the environment repository.	2 Minutes
5	Deploy Scripts	In TARGET environment deploy all scripts using the EMSE Tool.	5 Minutes
6	Update INCLUDES_CUSTOM_GLOBALS	In TARGET environment update the INCLUDES_CUSTOM_GLOBALS script as needed for this environment (debugging options, environment variable, etc.)	2 Minutes
7	Smoke Test	In TARGET environment perform smoke testing to ensure proper script execution.	10 Minutes