

2008

Scripting: Beyond the Basics

Seth Axthelm, Consulting Manager
Accela, Inc.

Overview

- Quick EMSE Overview
- Variable Branching
- Exercises
 - Setting Up Variable Branching
 - GIS scripting
 - ASI Tables
 - Complex Fees

What is EMSE?

- EMSE = Event Manager and Script Engine
 - Core Accela Automation® feature
 - Work together to allow customization to Accela Automation
 - Event Manager controls timing
 - Script Engine executes actions

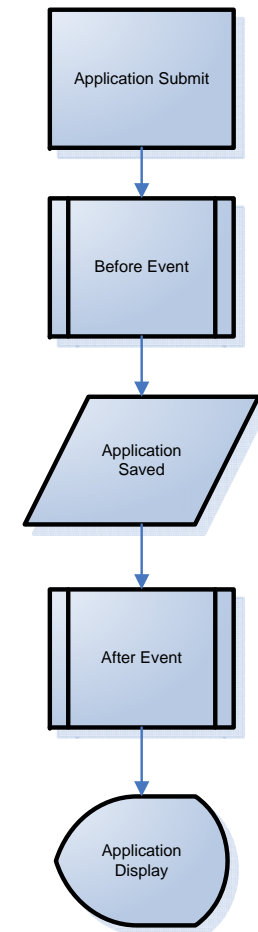
What is an Event?

- Action triggered by a user
- “Submit” button on a form
- Saving data
- 76 unique events defined in Accela Automation (20 added from last year)

The screenshot shows a web application interface for 'Task Details - Application Submittal'. At the top, it displays 'CAP ID: 07BRA-00000-00005'. Below this is a toolbar with icons for navigation and actions. The main form area is divided into several sections: 'Department' (BUILDING INSPECTION), 'Staff' (Clark Kent), 'Status' (Accepted w/Review Fee Paid), 'Status Date' (06/13/2007), 'Due Date' (06/12/2007), 'Comments' (need additional review please), 'Start Time', 'End Time', 'Billable' (checkbox), 'Overtime' (checkbox), and 'Hours Spent'. The 'Workflow' tab is selected, and the 'Workflow History' tab is also visible. The form is titled 'Task Details - Application Submittal'.

Event Types

- **Before**
 - Prior to database action
 - Database action can be cancelled via script
 - Typically used for validation
 - Example: ApplicationSubmitBefore
- **After**
 - After database action completes
 - Typically used for automation
 - Example: ApplicationSubmitAfter



What is a Script?

- What is a **Script**?
 - JavaScript code
 - Stored in the database
 - In order to be executed, must be attached to:
 - Event,
 - Batch Job, or
 - Script Test

```
function addFee(fcode,fsched,fperiod,fqty,finvoice) // Adds a single fee
(
    assessFeeResult = aa.finance.createFeeItem(capId,fsched,fcode,fperiod,fqty);
    if (assessFeeResult.getSuccess())
    {
        feeSeq = assessFeeResult.getOutput();
        message+="Successfully added Fee " + fcode + ", Qty " + fqty + br;
        debug+="The assessed fee Sequence Number " + feeSeq + br;
        if (finvoice == "Y")
        {
            feeSeqList.push(feeSeq);
            paymentPeriodList.push(fperiod);
        }
    }
    else
    {
        debug+= "ERROR: assessing fee (" + fcode + "): " + assessFeeResult.getErrorMessage() + br;
    }
}
```


Master Scripts

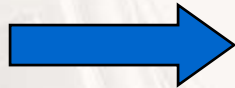
- Standardized scripts
- One script per event
- *Standard Choice* entries control the script
- Modular, plug-in functions
- Minimal JavaScript knowledge required

Events with Available Master Scripts

ApplicationConditionAddAfter	InspectionScheduleAfter
ApplicationSpecificInfoUpdateAfter	InspectionScheduleBefore
ApplicationSpecificInfoUpdateBefore	InvoiceFeeAfter
ApplicationStatusUpdateAfter	LicProfUpdateBefore
ApplicationSubmitAfter	ParcelAddAfter
ApplicationSubmitBefore	ParcelUpdateAfter
ContactAddAfter	PaymentReceiveAfter
ContactAddBefore	PaymentReceiveBefore
ContactEditAfter	RenewalInfoUpdateAfter
ContactRemoveAfter	V360InspectionResultSubmitAfter
FeeAssessAfter	WorkflowTaskUpdateAfter
FeeAssessBefore	WorkflowTaskUpdateBefore
InspectionResultSubmitAfter	

Master Script Flow Execution

Event



Master
Script



Script
Controls

CAP ID: 07BCA-00000-00005

ee Inspections Owner Parcel

Task Details - Application Submittal

Assigned Date
06/27/2007

Assigned to Department
Building

Current Status

City of Metroville

User ID:

Agency Profile User Profile Attachments Application Ped

Events - Event List

Edit	Event	Associated Script
	ApplicationSubmitAfter	ApplicationSubmitAfterV1.4

Standard Choices Item Name: [ApplicationSubmitAfter](#)

Description:
(250 char max)

Status: ☒ Enable ☐ Disable

Standard Choices Value	Value Desc
01	true ^ showDebug=true; showMessage=true;
02	appMatch("Licenses/*/*") ^ branch("LICENSE FEES")
03	appMatch("Building/Commercial/New/*") ^ branch("Update Impact Fees")
04	appMatch("Building/Residential/New/*") ^ branch("Update Impact Fees")

Script Control Design

- Sample Script Controls

Standard Choices Item - Edit

Use this form to set up a Standard Choices Item.

Standard Choices Item Name: APPLICATIONSUBMITAFTER

Description: (250 char max) Main script control for ApplicationSubmitAfter master script

Status: ☒ Enable ☐ Disable

Script Action Item matches value in related Master Script

order of execution

Script Control setting logging options

Standard Choices Value	Value Desc	Active
01	true ^ showDebug = true; showMessage = true	<input checked="" type="checkbox"/> Delete
02	appMatch("Building/Reroof/NA/NA") ^ addAllFees("BLDCR05",	<input checked="" type="checkbox"/> Delete

Script Control with Criteria and Action

Enable or Disable this Script Control

Update Add Cancel

h?fuseaction=RbizdomainSearch

Script Control Basics

- Script Controls
 - Are case-sensitive!
 - Do all the work. Criteria tells when, Action tells what
 - Criteria and Actions separated by a caret (^)
 - Must be valid JavaScript

Script Control Basics

- Use curly braces "{" and "}" to reference
 - Application Specific Info fields
 - Task Specific Info fields
 - Parcel custom attributes
- User "" around strings
- Standard Choices must be numbered sequentially

Script Control Structure

separator (^)

Criteria

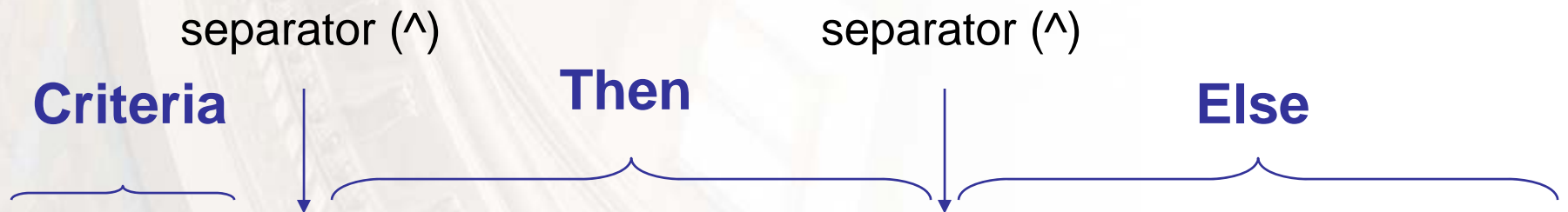
Action

`appMatch("Building/Reroof/NA/NA") ^ addAllFees("BLDCR05","FINAL",1,"N")`

IF this is true,

THEN do this

Script Control Structure - Else



```
{Acres Disturbed} < 5 ^ addFee("SMALLACRE","FEESCHED","BLD",1,"N")^ addFee("BIGACRE","FEESCHED","BLD",2,"N");
```

IF this is
true,

THEN do this

Else do this

Master Scripts Criteria

- **Criteria**
 - Must evaluate to only "True" or "False"
 - May contain multiple statements, separated by logical operators

Master Scripts Criteria Example

- true
- appMatch("Building/*/*/*")
- inspType == "Final Inspection"
- {Government Funded} == "Yes"
- {Building Type} != "Historical"
- {ParcelAttribute.Neighborhood} == "Downtown Area"
- feeExists("LICFEE") || balanceDue == 0
- inspType == "Final" && !isScheduled("Electrical")
- proximity("GIS","Schools",parseInt({Number of feet}))
- taskStatus("License Issuance","Issued")

Master Scripts Actions

- Action
 - Usually call functions developed for common actions
 - May contain multiple statements (separated by semi-colon)
 - addFee(); scheduleInspection(); etc...
 - Can span multiple lines
 - true ^ addFee()
 - ^ scheduleInspection();
 - ^ UpdateAppStatus();
 - May call subroutines using the branch() function

Master Scripts Action Examples

- activateTask("Plan Review")
- addFees("RENEW","LICENSE","PER2005",1,"Y",childApp)
- addAppCondition("Permit","Applied","Reinspection Fee","Re-Inspection Fee","Hold")
- closeTask("Peer Review","NA","Closed by Script","Closed by Script")
- comment("Square footage is " + {Sq Ft})
- childApp = createChild("Building","Commercial","Plumb","NA","")
- editAppSpecific("Total Value",{Sq Ft} * {Price per Sq Ft})
- branch("CALCVALUES")

Master Scripts Global Variables

- Script Global Variables
 - Event dependent
 - Workflow: wfTask, wfStatus, wfDate, etc.
 - Inspection Result: inspType, inspResult, InspGroup, etc.
 - Debug pop-up window shows the full list

Master Scripts Debug Window

Accela Automation: START - R9999-A - Windows Internet Explorer
http://aa.suppl.accela.com/EventManager/index.cfm?FUSEACTION=START

EMSE Script Results for 07BC00293

capId = class com.accela.aa.aamain.cap.CapIDModel
cap = class com.accela.aa.emse.dom.CapScriptModel
currentUserID = ADMIN
currentUserGroup = BuildingAdmin
systemUserObj = class com.accela.aa.aamain.people.SysUserModel
appTypeString = Building/Residential/Addition/Wall
capName = null
capStatus = Active
fileDate = 5/21/2007
fileDateYYYYMMDD = 2007-05-21
sysDate = class com.accela.aa.emse.dom.ScriptDateTime
parcelArea = 0
estValue = 15000
calcValue = 0
feeFactor = CONT
capDetail = class com.accela.aa.emse.dom.CapDetailScriptModel
houseCount = 1
feesInvoicedTotal = 0
balanceDue = 0
wfTask = Application Submittal
wfStatus = Approved
wfDate = 2007-06-21
wfDateMMDDYYYY = 06/21/2007
wfStep = 1
wfComment = null
wfProcess = BCRESADD
wfNote = null
Executing: PreExecuteForAfterEvents, Elapsed Time: 0.703 Second
Finished: PreExecuteForAfterEvents, Elapsed Time: 0.703 Second
{Engineered Elevation} = null
{Pipe Size} = null
{Storm Damage} = No

CAP that triggered the event

Pre-populated CAP values available to your script

Event specific values

User-defined data (ASI, TSI, Parcel attributes, etc.)

Done

Master Scripts Branching

Standard Choices Item - Edit

Use this form to set up a Standard Choices Item.

Standard Choices Item Name: Application Submittal Script Control

Description:
(250 char max)

Status: ☒ Enable ☐ Disable

Standard Choices Value	Value Desc	Active
01	appMatch("Building/**/*") ^ branch("CALC VALUES")	<input type="checkbox"/> Delete

Standard Choices Item - Edit

Use this form to set up a Standard Choices Item.

Standard Choices Item Name: CALCVALUES

Description:
(250 char max)

Status: ☒ Enable ☐ Disable

Standard Choices Value	Value Desc	Active
01	true ^ IBCValue = lookup("IBC VALUES",{IBC Group Code})	<input checked="" type="checkbox"/> Delete
02	{MultiUse Value Heated} > 0 ^ editAppSpecific("Heated Va	<input checked="" type="checkbox"/> Delete

Variable Branching Overview

- In Master Script versions prior to 1.4, branches needed to be hard coded:

true ^ branch("Assess Fees")

- As of Master Script version 1.5, new feature Variable Branching allows us to create the branch destination using variables:

true ^ branch(variableName1 +variableName2)

Turning on Variable Branching

- To make use of this functionality the following flag needs to be set to true in each script you wish to use it in:

var enableVariableBranching = true;

- **Note:** when using variable branching, the documentation functionality of the scripts no longer works

So What Does This Mean?

- Many uses for variable branching
- Use the CAP type as a part of the variable branching.
- Allows us to put the following script controls for an event (ie. ApplicationSubmitAfter)

```
true ^ branch("ASA:" + appTypeArray[0] + "/*/*/*")  
true ^ branch("ASA:" + appTypeArray[0] + "/" + appTypeArray[1] + "/*/*")  
true ^ branch("ASA:" + appTypeArray[0] + "/" + appTypeArray[1] + "/" +  
appTypeArray[2] + "/*")  
true ^ branch("ASA:" + appTypeString)
```

So What Does This Mean?

- Which for CAP Type, “Permits/Drainage/Non-Residential/NA”, would, based on the code above, branch to the following Standard Choices:

ASA:Permits/*/*/*

ASA:Permits/Drainage/*/*

ASA:Permits/Drainage/Non-Residential/*

ASA:Permits/Drainage/Non-Residential/NA

So What Does That Mean?

- Put your scripts based on where they need to be called from

ASA:Permits/*/*/*

(Any script controls put in this Standard Choice would be executed for all CAP types in the Permits module)

ASA:Permits/Drainage/*/*

(Any script controls put in this Standard Choice would be executed for any CAP types that fall under the Permits/Drainage/ type)

ASA:Permits/Drainage/Non-Residential/*

(Any script controls put in this Standard Choice would be executed for any CAP types that fall under the Permits/Drainage/Residential/ sub-type)

ASA:Permits/Drainage/Non-Residential/NA

(Any script controls put in this Standard Choice would be executed for any CAP types that fall under the Permits/Drainage/Non-Residential/NA cap type)

What About Other Events?

- We have seen that the Standard Choice before was prefixed by “ASA” standing for ApplicationSubmitAfter. We hard coded that in the Standard Choice for that event.
- In other events you will want to hard code a different prefix for that event’s Standard Choice.

What About Other Events?

- For example, for the WorkflowTaskUpdateAfter event you would have the following entries:

true ^ branch("WTUA:" + appTypeArray[0] + "/*/*/*")

true ^ branch("WTUA:" + appTypeArray[0] + "/" + appTypeArray[1] + "/*/*")

true ^ branch("WTUA:" + appTypeArray[0] + "/" + appTypeArray[1] + "/" + appTypeArray[2] + "/*")

true ^ branch("WTUA:" + appTypeString)

- It is the same except for the prefix that is entered.

What About Other Events?

- Abbreviations used as prefixes for each of the events:

Event	Abbrev	Event	Abbrev
ApplicationConditionAddAfter	ACAA	InspectionScheduleAfter	ISA
ApplicationSpecificInfoUpdateAfter	ASIUA	InspectionScheduleBefore	ISB
ApplicationSpecificInfoUpdateBefore	ASIUB	InvoiceFeeAfter	IFA
ApplicationStatusUpdateAfter	ASUA	LicProfUpdateBefore	LPUB
ApplicationSubmitAfter	ASA	ParcelAddAfter	PAA
ApplicationSubmitBefore	ASB	ParcelUpdateAfter	PUA
ContactAddAfter	CAA	PaymentReceiveAfter	PRA
ContactAddBefore	CAB	PaymentReceiveBefore	PRB
ContactEditAfter	CEA	RenewalInfoUpdateAfter	RIUA
ContactRemoveAfter	CRA	V360InspectionResultSubmitAfter	VIRSA
FeeAssessAfter	FAA	WorkflowTaskUpdateAfter	WTUA
FeeAssessBefore	FAB	WorkflowTaskUpdateBefore	WTUB
InspectionResultSubmitAfter	IRSA		

Exercises

- Handouts
- Exercises
 - Setting up variable branching
 - GIS scripting
 - ASI Table interactions
 - Inter-cap scripting
 - Complex fees

Questions?

Seth Axthelm

saxthelm@accela.com