



Event Manager and Script Engine

Source Code Control Systems

2015



Example Requirements

- Accela Automation version 7.3.3.3
- Eclipse IDE with SVN plugins installed
- Assembla Account with Access to the Accela Certification Training Repository



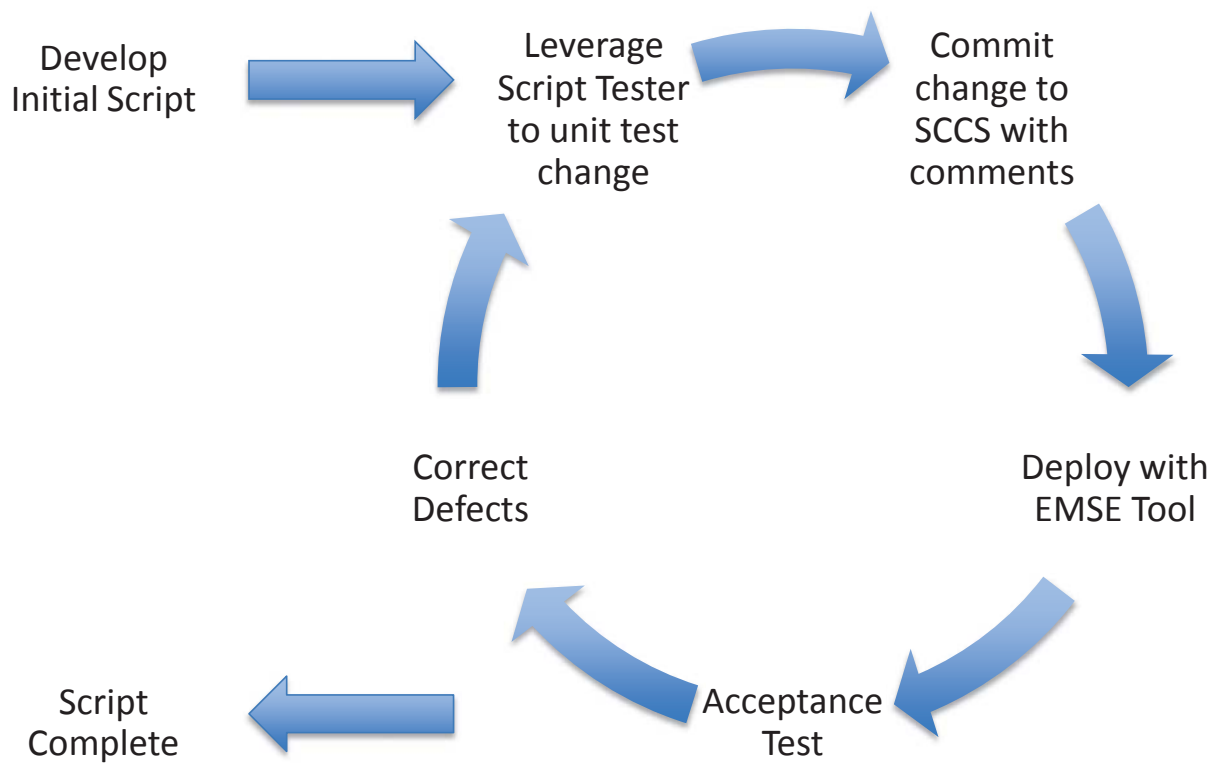
What you will learn

- Linking a Source Code Control System (SCCS) to Accela Automation
 - An SVN SCCS will be used for this example and is the repository type used by Accela Professional Services
- Basic SVN Operations
 - Check Out
 - Commit



The Development Cycle

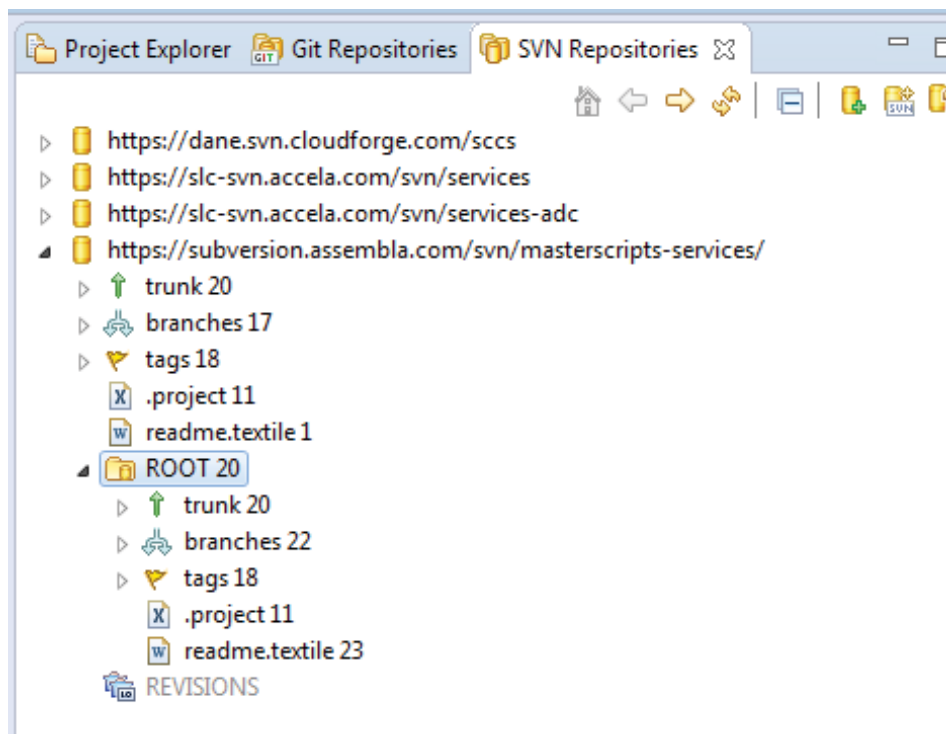




Looking at the Setup



Eclipse IDE integrated to SVN



SVN Repository in Assembla






[Assembla](#) > [My Start](#) > [MasterScripts-Services](#)

MasterScripts-Services

[Team](#) [Stream](#) [SVN](#)

[Source](#) [Commits](#) [Compare](#) [Merge Requests](#) [Instructions](#)

masterscripts-services

NAME	DATE	COMMIT MESSAGE
 branches	2015-02-17	[Dane Quatacker]
 tags	2015-02-17	[Dane Quatacker] Beta Release of Master Scripts
 trunk	2015-02-17	[Dane Quatacker] Correct parameter in ScheduleInspect
 .project	2015-02-17	[Dane Quatacker]
 readme.textile	2015-02-17	[Dane Quatacker] Added repository Readme

readme.textile

Services Master Script Repository

This file is designed to give a brief overview of the directory structure used in this SVN repository

>Root

readme.textile – This file

>trunk – Active branch. Do not commit anything to branch that has not been unit tested

>Documentation – Documentation folder containing information on usage of master scripts

>INCLUDES_ACCELA_FUNCTIONS – Folder containing the accelera includes functions each function should be its own .js file

>INCLUDES_ACCELA_FUNCTIONS_ASB – Folder containing the accelera includes for application submit before each function should be its own .js file

>Master Scripts – Folder containing the master scripts heads that get associated to an event

>Misc – Folder containing Data Manager export and various utilities not directly tied into Master Script framework

>Sample Repository Structure – Folder and files for seeding a Agency Repository

>Templates – Script files that are used as primers for common actions, but are not part of the master script framework

>tags – Contains release information. No new features should be added to tags. After applying bug fix to a release ensure this is merged into Trunk


> BETA 3.0.9619 – current release that is done on community

>branches – When work is to be done, create a branch for feature being created. After unit testing the branch merge back into trunk and remove.

SVN Repository in EMSE Tool (Using the Trunk)



Accela Configuration



City of Metroville

Menu | Favorites | Help | Logout | Building

User ID: ADMIN

Admin Tools | Daily

ACCELA AUTOMATION®

Agency Profile | User Profile | Attachments | Application | People | Property | Fees | Inspection | Condition | Workflow | Calendar | Events

Standard Choices Item - Edit

Use this form to set up a Standard Choices Item.

Standard Choices Item Name: EMSEToolConfig

Description: (250 char max)

Status: ☒ Enable ☐ Disable

Type: ☒ System Switch ☐ Shared drop-down ☐ EMSE ☐ Business Configuration

Standard Choices Value	Value Desc	Active	
agency_repo_password	password	<input checked="" type="checkbox"/>	Delete
agency_repo_username	quatacker	<input checked="" type="checkbox"/>	Delete
agency_url_svn	http://subversion.assembla.com/svn/agencyimpref/trunk	<input checked="" type="checkbox"/>	Delete
master_repo_password	password	<input checked="" type="checkbox"/>	Delete
master_repo_username	quatacker	<input checked="" type="checkbox"/>	Delete
master_url_svn	http://subversion.assembla.com/svn/services-emse/trunk	<input checked="" type="checkbox"/>	Delete

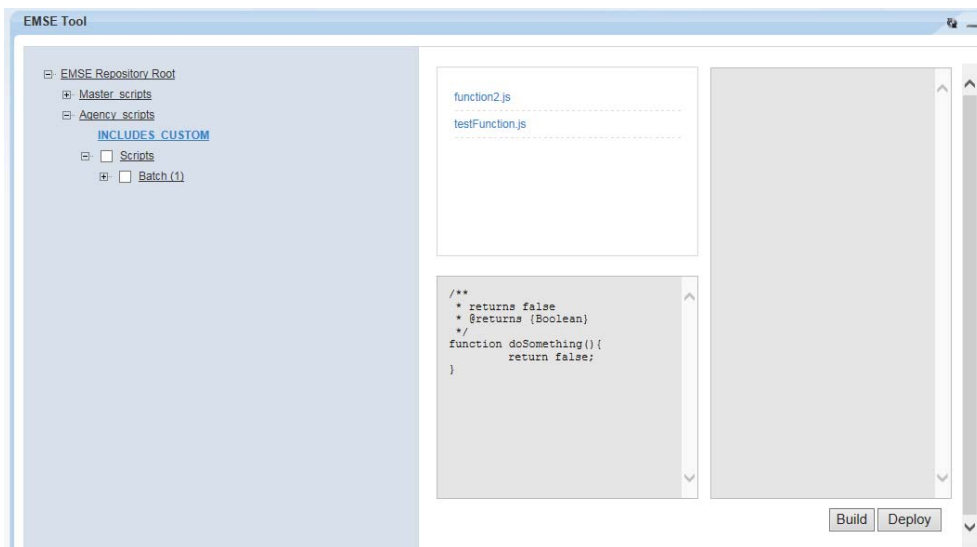
Update | Add | Cancel

INCLUDES_CUSTOM



Includes Custom

- Folder in repository should be at root and named INCLUDES_CUSTOM
- Each file in folder should represent a **SINGLE FUNCTION**
- EMSETool can compile into single file and deploy automatically to Accela



- Clicking Build will compile the file and minify the JavaScript
- Clicking Deploy will send to Accela

Includes Custom in Accela

Agency Profile	User Profile	Attachments	Application	People	Property	Fees	Inspection	Condition	Workflo
----------------	--------------	-------------	-------------	--------	----------	------	------------	-----------	---------

Custom Script - Custom Script Detail

Master Script Name: INCLUDES_CUSTOM

Master Script Text:

```
function doSomething(){return false}function testMessage(vmsg){aa.print(vmsg)};
```

Last Modified Date: 1/27/2015 by ADMIN

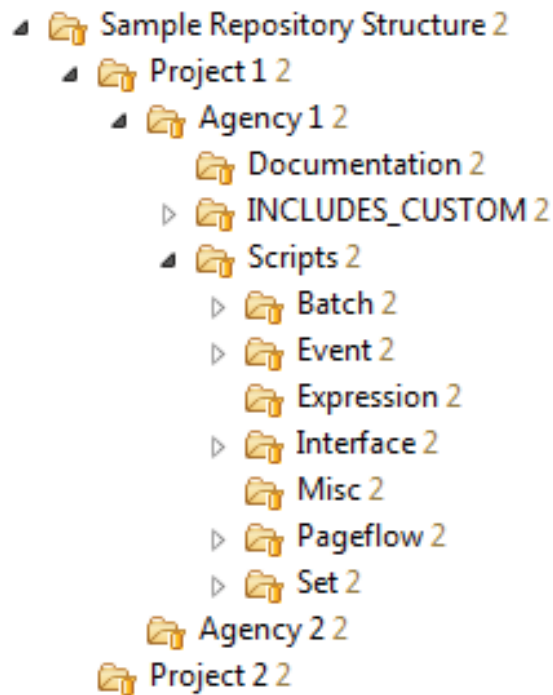
Save

Repository Structure and Naming



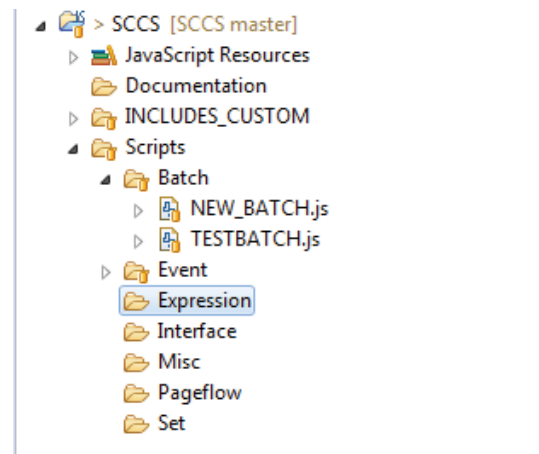
Best Practices for Repository Structure

- Only one source control repository can be linked to Accela at a time
- Best practices for structure are provided in EMSE 3.0 distribution



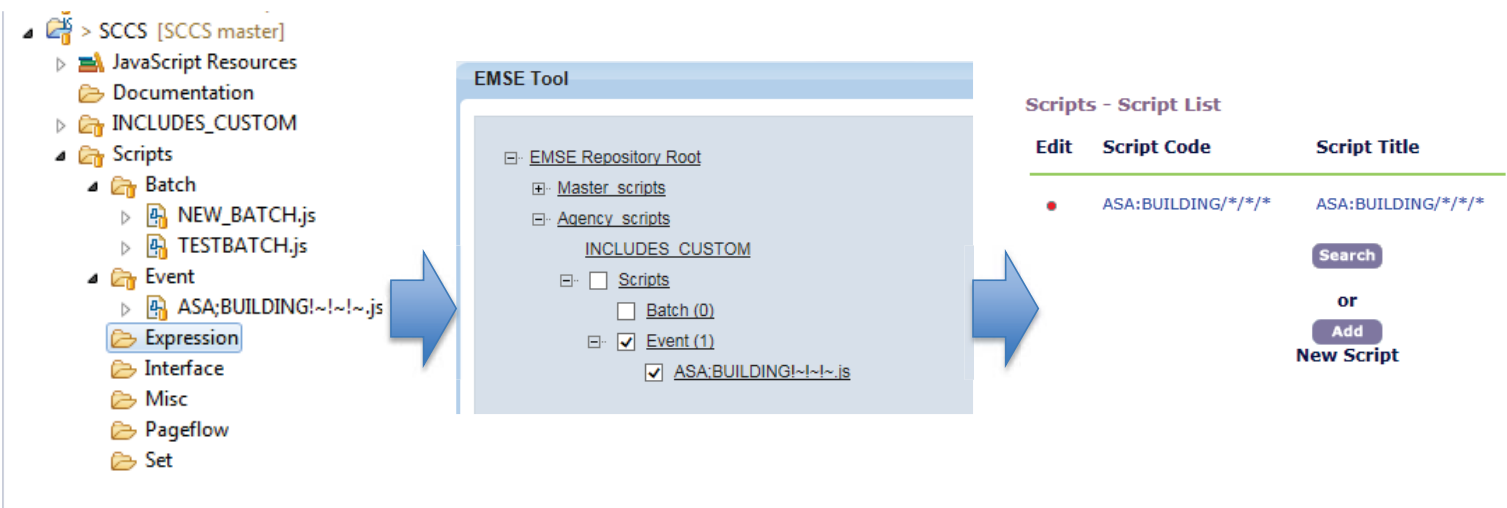
Script Naming

- Function in the INCLUDES_CUSTOM folder should be camelCased
 - Ex. getApplicationSpecific(), getCapId(), setUserId()
- Scripts source in the Scripts folder should be all uppercase with a .js extension.
 - Ex. TESTBATCH.js



Script Naming Continued

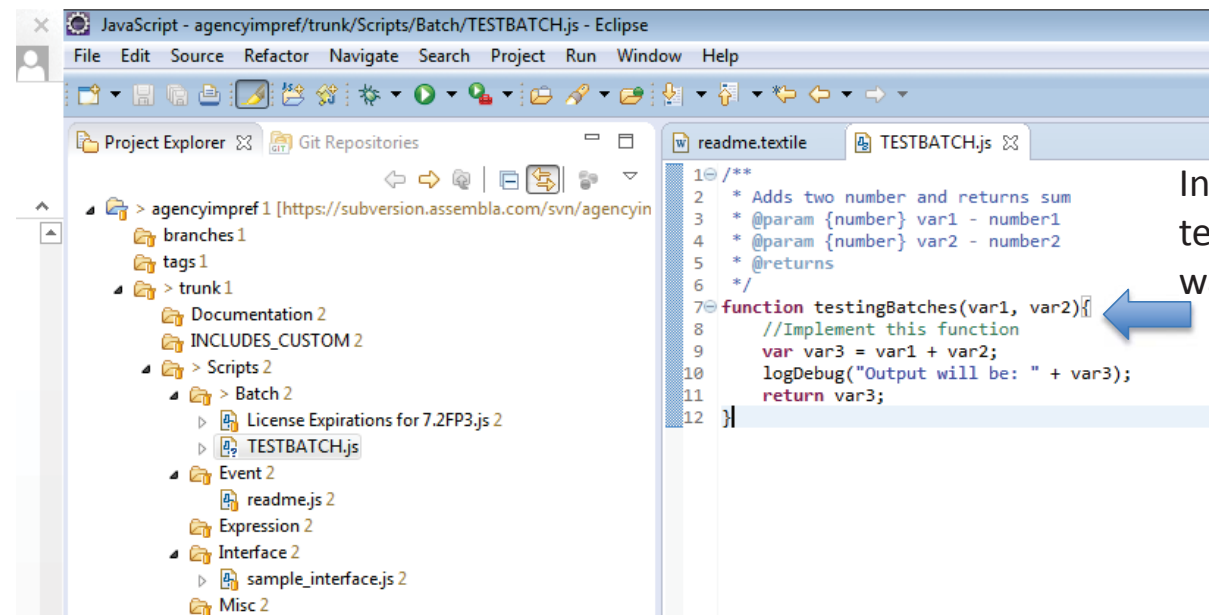
- Event scripts should be all uppercase and use proper token replacement
 - ; = : , ~ = * , ! = /
 - Ex. ASA;BUILDING!~!~!~.js = ASA:BUILDING/*/*/*



A walk through the process!



Modify in the Script in Editor



In this example the testingBatches function was updated

Test the Script Using Script Test

The screenshot displays the 'Script Test' window in the Accela Automation application. The window is titled 'Accela Automation: START - R9999-A - Internet Explorer'. It shows the results of a script execution for the event 'ApplicationSpecificInfoUpdateAfter'.

Script Test Details:

- Warning:** Improperly written scripts may incorrectly alter data for many records.
- Script Transaction:** Always Rollback (dropdown menu)
- Script Initializer:** (empty text area)
- Script Text:**

```
var myCapId = "LPV-000032-0304";
var myUserId = "ADMIN"

var eventName = "ApplicationSpecificInfoUpdateAfter";
aa.env.setValue("EventName",eventName);
//wfTask = "Equipment Status";
//wfStatus = "In Service";

var controlString = eventName;

var runEvent = true; // set to false if you want to roll your own code here in script
/* master script code don't touch */ var tmpID = aa.cap.getCapID(myCapId).getOut
```

Execution Output:

- EMSE Script Framework Versions
- EVENT TRIGGERED: ApplicationSpecificInfoUpdateAfter
- SCRIPT EXECUTED:
- INCLUDE VERSION: 3
- SCRIPT VERSION : 3
- GLOBAL VERSION : 3
- lookup(EMSE_VARIABLE_BRANCH_PREFIX,ApplicationSpecificInfoUpdateAfter) = ASIUA
- Output will be: 3

Event Manager Message:

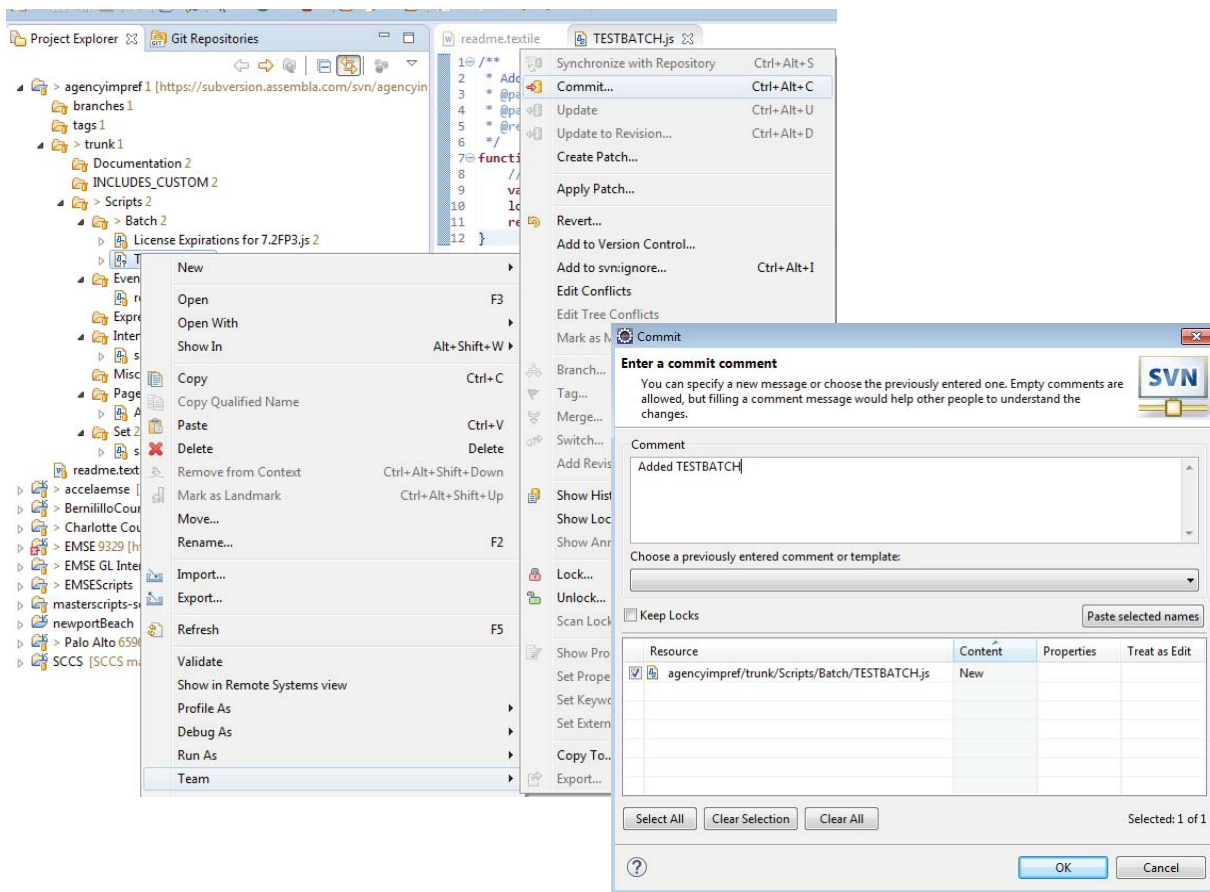
- There is 1 new message from the Event Manager.
- Close button

Additional Information:

- Switch to [Advance](#) mode.
- Version 7.3.3.3.0_4419
- 100% zoom level

A blue arrow points from the text 'Test using the ScriptTest' to the 'Output will be: 3' line in the execution output.

Commit to SVN



Seeing Changes in EMSE Tool and Deploy

EMSE Tool

[-] EMSE Repository Root

[-] Master_scripts

[-] Agency_scripts

INCLUDES_CUSTOM

[-] ☒ Scripts

[-] ☒ Batch (1)

☒ TESTBATCH.js

If script is new you must right click the script and Deploy, the button will not be available

Number of differences: 14 differences from 9 lines of code.

Source Code Control System Script

(Modified by dqatacker at 2015-01-27 09:11)

1

2

3

4

5

6

7

8

9

10

11

12

/**

* @dis t/c numbers and returns sum

* @param var1 {number}

* @param var2 {number}

* @returns {number}

*/

function testingBatches (var1, var2)

AA Script

1

2

3

4

5

6

7

8

9

10

11

12

/**

* @returns false every time!

* @returns {Boolean}

*/

function testingBatches () {

return false;

}

Clicking the Deploy will sent to Accela

Deploy

The script [TESTBATCH] is deployed from Source Code Control System to AA successfully.

The Script in Accela Automation

Scripts - Script Detail

Script Code: TESTBATCH

Script Title: TESTBATCH

Script_INITIALIZER:

Script Text:

```
/**
 * Adds two numbers and returns sum
 * @param var1 {number}
 * @param var2 {number}
 * @returns {number}
 */
function testingBatches(var1, var2){
    //Implemented this function
    var var3 = var1 + var2
    logDebug("Output will be: " + var3);
    return var3;
}
```

Last Modified Date: 1/27/2015 by ADMIN

Save

Script Test

Delete