

Configuring the Universal Script

Description

Configuring the Universal Script

When you create a new script file, you copy the contents of the UniversalMasterScript file into your new script file ([Creating a New Script](#)). You then need to modify this copied content to configure it for your particular application.

To configure the universal script

1. Locate the `START Configurable Parameters` section of the master script.
2. Set the value of the `controlString` variable to the name of the Standard Choice. The Standard Choice name must match the name of the event for which the Standard Choice contains the script controls.

```
var controlString = "<Standard
Choice>"
```

3. Set the value of the `preExecute` variable to indicate whether to trigger the script before or after the event.

```
var preExecute = "<before
or after>"
```

where:

`<before or after>` is `PreExecuteForBeforeEvents` for before events and `PreExecuteForAfterEvents` for after events.

4. Set the `documentOnly` variable to specify whether or not to display the hierarchy of standard choice steps.

```
var documentOnly = false
```

5. Configure the remaining sections as required.

- The following section of the master script configures the internal version of the master script file and the global master scripts to include during runtime.

```
var SCRIPT_VERSION = 2.0
eval(getScriptText("INCLUDES_ACCELA_FUNCTIONS"));
eval(getScriptText("INCLUDES_ACCELA_GLOBALS"));
eval(getScriptText("INCLUDES_CUSTOM"));
```

Note:

The ApplicationSubmitBefore event includes the INCLUDES_ACCELA_FUNCTIONS_ASB master script instead of the INCLUDES_ACCELA_FUNCTIONS master script

Note:

The master script files include the INCLUDES_CUSTOM master script as a placeholder to incorporate customizations to the master script. Civic Platform does not provide the INCLUDES_CUSTOM master script so as not to overwrite existing master script customizations during system upgrades.

- This section includes the scripting to evaluate the value of the `documentOnly` variable configured in the previous section.

```
if (documentOnly) {
  doStandardChoiceActions(controlString,false,0);
  aa.env.setValue("ScriptReturnCode", "0");
  aa.env.setValue("ScriptReturnMessage", "Documentation Successful. No actions executed.");
  aa.abortScript();
}
```

- The `BEGIN Event Specific Variables` section loads the values for the variables of the associated event ([AdditionalInfoUpdateAfter Variables](#)).

Figure: AdditionalInfoUpdateAfter Variables

Events - Event Detail**Event Name:** AdditionalInfoUpdateAfter**Description:** This event occurs after a Additional Info is updated.**Script Name:** **Last Modified Date:** 5/23/2012 by ADMIN**Save** this event.**Delete** disable event.**List of Input/Output Environment Variables:**

- IN: AdditionalInfoBuildingCount
- IN: AdditionalInfoConstructionTypeCode
- IN: AdditionalInfoHouseCount
- IN: AdditionalInfoPublicOwnedFlag
- IN: AdditionalInfoValuation
- IN: CurrentUserID
- IN: PermitId1
- IN: PermitId2
- IN: PermitId3

For example, Civic Platform uses the following variables for the AdditionalInfoUpdateAfter event.

```
var aiBuildingCount = aa.env.getValue("AdditionalInfoBuildingCount");
var aiConstructionTypeCode = aa.env.getValue("AdditionalInfoConstructionTypeCode");
var aiHouseCount = aa.env.getValue("AdditionalInfoHouseCount");
var aiPublicOwnedFlag = aa.env.getValue("AdditionalInfoPublicOwnedFlag");
var aiValuation = aa.env.getValue("AdditionalInfoValuation");
This variable list corresponds to the default set of variables defined for the event.
```

Note:

The INCLUDES_ACCELA_FUNCTIONS master script resolves the CurrentUserID, PermitId1, PermitId2, and PermitId3 global variables.

- After logging event specific variable, the master script executes the `Main Loop` by performing the actions prescribed by the applicable Standard Choice script controls.

```
doStandardChoiceActions(controlString,true,0);
```

Version Number All Versions

Products Accele Automation