

AcroTeX.Net

## AcroTeX PDF Blog

# Loading and Saving Widget Settings

D. P. Story

The source files for this AcroTeX PDF Blog are attached to this PDF.  
Click [blog11](#) and save this file as `blog11.zip`.

## 1. Introduction

Joel Gernaci, Technology Guru for Acrobat, in a recent article titled *Saving the State of SWF Content in Acrobat 9*, introduces two *undocumented* methods,

```
multimedia_saveSettingsString  
multimedia_loadSettingsString
```

for saving and loading values of SWF variables. These “multimedia” functions are called by the `ExternalInterface.call` method.

In the article, he illustrates the use of these two methods, and provides the MXML source file of his example. What is not clear in his comments from within the source code is how multiple values are saved and loaded using these methods, and whether these are global methods, allowing essentially for inter-widget communication. The purpose of this article is to explore both of these questions.

## 2. Load and Save Values

After many hours of pain and agony, and beating my head up against the keyboard, I have managed to develop some techniques for saving and retrieving multiple values, and have determined that loading and saving is local to each widget.

Below are two RMA (rich media annotations) each one using `widget1.swf`. My example is not nearly so colorful as Joel’s, but the purpose was to develop techniques, not to solicit the admiration of my fellow readers.



Enter text into any or all of the text fields of RMA 1 or RMA 2. (Be sure you “commit” the data, much as you must do with Acrobat form fields.) These new values are saved using `multimedia_saveSettingsString`. Page to the title page then back again, when the widgets are activated, `multimedia_loadSettingsString` is called and the values you entered populate the text fields. If you are using Acrobat and viewing this document outside a browser, you can save and close the document. When you open the document again, the values you entered are still there.

- Note that the RMAs both use `widget1.swf` but the data for each are save separately. So, the saving and the loading of data has a local context.

### 3. The MXML Code

Below is a partial listing of `widget1.mxml`. Comments on the code follow.

We store the values of the two `TextInput` fields as values of the XML List, `flashSettings`. Originally, I tried using an object to hold the field data, but could not make it work. I settled on using XML techniques.

**initApp():** When the application is activated, the `initApp()` function executes. In lines 14–15, we call `multimedia_loadSettingsString`, we set the return value type `String`.<sup>1</sup>

- If `cResult` is not the empty string (or null), we create a new XML variable called `flashSettings`, lines 23–24. We initialize the values for `flashSettings.settings.text1` and `flashSettings.settings.text2`. The Boolean `bProcessValueCommit` is used to turn off the `valueCommit` processing defined for each of the `TextInput` fields as we update the values of these fields. We then convert `flashSettings` to an XML string in line 31, then call `multimedia_saveSettingsString`.
- If `cResult` is non-empty (hence `flashSettings` must have already been defined), we load in `cResult` into `flashSettings` by

```
flashSettings = new XML(cResult);
```

in line 17. We then set the values of the two `TextInput` fields in lines 18–21.

**onValueCommit():** The `TextInput` fields have a `valueCommit` property. When the value is changed (committed), the function `onValueCommit()` is called. This function performs actions similar to `initApp()`, it updates the values of `text1` and `text2` elements, lines 38–39, convert to an XML string (line 40) and save this little dude using `multimedia_saveSettingsString`.

It works!

---

<sup>1</sup>Joel typed the return value as `Object`, `String` seems to work better.

#### 4. Finally

If you want a “reset” button, you can build one using an Acrobat button that calls a function in the widgets to reset their field to their initial values. This is not implemented in this example and is left to an exercise for you.

Now, I simply must get back to my retirement! ☹