

## Section IV

### 5

# Delivery

During the initial demonstration of [AllWayallWay](#), we ~~had~~ set up the application on Rash Decisions Advertising's Linux server. Further, Flash Communication Server MX ~~had been was~~ set up in the [Production Mode](#) [\[\[Is this supposed to be capped?\]\]](#), so we ~~did not don't~~ have to reinstall it. ~~So Thus~~, staging the final application ~~was is going to be~~ easy—; all we need ~~ed~~ to do ~~was is~~ place the application file pairs (host/guest) into the application folder and re-install the main.asc file in the special folder for the server-side files required in the [Production mode](#) [\[\[Here, "Production" is capped but "mode" is lowercased...\]\]](#).

We also want ~~ed~~ to spend a little more time with RDA's IT consultant. If the software need ~~ed~~ tweaking, we want ~~ed~~ him to be able to easily deal with re-installation, updates, or other issues that might arise. The Comm Server is able to listen to different ports, and we want ~~ed~~ to make sure that nothing in the security system he ~~had~~ set up for Karen ~~would will~~ conflict with the ports used. Also, if we spend ~~t~~ some time going over the way Flash Communication Server MX work ~~ed~~, we ~~would will~~ get fewer calls for assistance in the future. Not that we don't support customer service—rather, we believe that the best customer service is when the customer is happy with the product and it works without problems.

## Creating the Module Pairs

Now that we ~~were are~~ ready to deliver the application, we ~~had have~~ to go back and create 14 separate FLA files. Each one ~~would will~~ be a host/guest combination. That ~~meant means~~ we ~~would will~~ generate 14 SWF files and 14 HTML files, for a [total of 42 files](#) [\[\[Isn't it 28 total files?\]\]](#). Adding the single main.asc file, we ~~had have~~ [43](#) [\[\[ "29" ? \]\]](#) files all together. All ~~would will~~ use the same application, but different files ~~would will~~ be needed so that all of the RDA crew ~~had members have~~ their own host/guest module sets of files. This process ~~concerned concerns~~ Jim.

<<Format as Memo>>

**Memo**

Date: ~~February 20~~

To: Bill and Nancy

From: Jim

Subject: A single file application

The strategy of creating asymmetrical files with video/audio communication seems to be wasteful. I realize that the files have important differences, like the ability to **set the limit of users** **[[Do you mean "set a limit on the number of users"??]]**, but that could be handled with a password for the "host" or some other way that could be done with a single file. We could also allow ~~the users~~ to create as many or as few video/audio windows as needed for a conference rather than limiting them to some set number. In fact, we've built several applications that generate new movie clips containing video/audio as needed by users joining a conference, so I'm at a loss as to why we've been creating applications with asymmetrical files.

Each of the users at RDA would be able to use the same file. All they'd need to coordinate ~~is~~ ~~would be~~ the time of a meeting. The entire project would take only ~~3~~ ~~three~~ files instead of ~~43~~ **[[?"29"??]]**! So next time we work on a project like this, let's consider a single-file approach and put the server-side **Stream** **[[Is this an object class??]]** object to work. **[[From this it sounds like you might develop a single-file application for a future project but not this one.]]**

<<Format as Memo>>

**Memo**

Date: ~~February 20~~

To: Jim and Nancy

From: Bill

Subject: A single file application

First of all, I like the idea of putting the server-side **stream** **[[?"Stream," as above??]]** object to better use. I can think of a lot of uses for it, especially if we want to have a little more sophisticated combination of streams. We ought to kick around some interesting uses for the **stream** **[[?"Stream," as above??]]** object at our next meeting.

As for the idea of single-file apps and any number of added users, that has as many drawbacks as it does advantages. These are my thoughts on the matter: **[[Production, please indent the bullet list]]**

- If we had a single-file app, we would still have to create seven applications for the system to work right. Each of the identified members of RDA would need ~~their~~ ~~his or her~~ own application ~~---~~ that would mean four files for each of the seven: ~~one~~ ~~4~~ FLA file, ~~one~~ ~~4~~ SWF file, ~~one~~ ~~4~~ HTML file, and a main.asc file. So, you've still got 28 files, plus an additional ~~7~~ ~~seven~~ folders. Each of the apps would need a unique name and RTMP reference. (If we used a single file that everyone could log on to, we could create separate areas for the different discussions, but then we'd have to make sure ~~that~~ everyone went to the right area at the right time, and mistakes would be far more common.)

☞ We don't ~~["want to"?!]~~ put the client into an embarrassing position ~~by-with~~ an application overload. We've all been involved in multi-user ~~a/v/v~~ sessions, and they can get herky-jerky as soon as the place fills up. Latency, freezes, and lost packets go sky-high, and even if ~~#the application~~ works, everyone is running on near-minimum quality. Where you have the mix of connection speeds in a company (not to mention the unknown connection speeds of clients), you need to build in a very good system where you can have communication without freezes and tie-ups.

It's one thing to push the envelope, and it's another to get too experimental with our clients. However, I agree that we should be doing a lot more with Server-Side Communication ActionScript, and I also think we should look into ways to optimize our apps. We need to see what can be done with ~~both~~ single-file ~~as well as multfile~~ apps, ~~as well as multi-file apps~~, both for optimizing the apps for clients and ~~for~~ our own work process.

As usual, Jim brings up important points to consider. However, the clients are paying the bills, and more than anything, we want to be sure that our applications serve the client's needs, with a minimum of muss and fuss for them. Too many problematic apps leads to disappointed clients, and we want to follow a path where safe is more important than slick. At the same time, we will push the envelope to discover the limits of an application. Such experimentation gives us new tools to meet client needs.

Making the host/guest module for Rash Decisions ~~Agency-Advertising only took~~ takes only an hour. Each module pair has a unique instance name in the RTMP address. For example, Karen Rashish's is,

```
"rtmp:/rashDec/CEO"
```

and June Le's is,

```
"rtmp:/rashDec/june"
```

We use the name preferences ~~that Karen Rashish~~ provided ~~by Karen Rashish~~, and employ ~~the same names them~~ in ~~namings~~ the file ~~names~~. The host module for Karen Rashish, for example, is CEOh.html and the audience module is CEO.html ~~(the host simply has an h added to the filename)~~. The same combination is used with all ~~of~~ the other file sets. ~~The host simply has an "h" added to the file name.~~

## Contact Menu

When people contact any of the personnel at Rash Decisions ~~Agency-Advertising~~, they don't want to have to fumble through a long URL ~~to contact the different people in the agency~~. To make it easy, we decided ~~to make a menu that would lists~~ all ~~of~~ the employees who have a host module.

Simply by placing FPushButton UI components on the stage, I ~~was-am~~ able to apply the same styles by copying the style code from the host/guest modules. Then, using the different names given to us by Karen Rashish, I ~~made-make~~ an array to set the click handlers for the buttons, as shown in the following segment:

```
//Set the click handlers
stuffURL = function () {
    _____urlNames = new Array(6);
```

```

_____ __urlNames.push("CEO", "jeb", "sheryl");
_____ __urlNames.push("kelley", "katisha", "brad", "june");
};
stuffURL();
for (var n = 6; n>=0; n--) {
_____ __urlLabel = eval("rash"+(n+1));
_____ __urlLabel.setClickHandler(urlNames.pop());
}

```

Next, each of the buttons has a function to identify the file-name of the module (the guest module) to load. **The following two** ~~[[The following two what?]]~~ show how the handler functions ~~were~~ are created:

```

//Handlers
CEO = function () {
_____ __rashName = "CEO.html";
_____ __findURL();
_____ __getURL(rashURL);
};
jeb = function () {
_____ __rashName = "jeb.html";
_____ __findURL();
_____ __getURL(rashURL);
};
//Remaining handlers

```

Finally, the script contains a function that identifies the path to the URL. This function is used by all ~~of~~ the button handlers:

```

//URL to Host module
findURL = function () {
_____ __rashURL = "http://www.rashDecisionsAdvertising.com/rashDec/"+rashName;
};

```

Using the same style created by Design Street, we integrate the menu ~~was integrated~~ as a gateway to the communication modules. Figure IV-5.1 shows what we created.

#### <<Figure IV-5-1.tif>>

Figure IV-5.1. ~~\_\_\_\_\_~~ The menu **[[“of Rash Decisions staff members’ names”?]]** simplifies the process of entering ~~the~~ conference meetings.

Because Design Street ~~had~~ incorporated the general look and feel of Rash Decisions Advertising’s site, the transition from their existing pages to the **menu** **[[Say what menu?]]** is virtually seamless. After installing the menu and application, I ~~gave~~ give Karen Rashish the URL for the menu, and ~~suggested~~ she contact the company that created her Web site and have them include a link to it where they ~~thought~~ think it’s most appropriate. In the meantime, she ~~could~~ can use the URL to the menu as the direct link, and I ~~gave~~ give her each of the URLs to the host HTML files so that ~~they~~ **[[“RDA”?]]** ~~could~~ can put them to use immediately. I ~~did~~ suggest that they not give out the host URLs lest someone begin using ~~them~~ the URLs for ~~their~~ his or her own communications.

We ~~were~~are paid the final installment for the job, ~~and~~ ~~we~~ ~~told~~ ~~tell~~ Karen ~~that~~ we ~~would~~ ~~will~~ be available ~~were they to if they~~ have any unforeseen problems with the applications. We also provided them with our own online video/audio communication URL, which we ~~were~~ now ~~including~~ in all ~~of~~ our business cards, stationery, and email signatures.

## Post~~m~~m-Mortem

-As usual, we gathered ~~ed~~ together to hash out what we could have done, what enhancements we might add, and how we might adopt the best features of the application for future applications. Jim and I agreed ~~d~~ not to rehash the single ~~vs~~versus multiple ~~file~~ issue. The following meeting notes provide an overview of what we ~~thought~~ think we would like to do and what possibly we should have done.

<<Format as Meeting Notes>>

### Meeting Notes

~~Date:~~ February 21

~~In-attendance:~~ Present: ~~\_\_\_\_\_~~ Nancy Durocher

- ~~\_\_\_\_\_~~ Jim Ford
- ~~\_\_\_\_\_~~ Bill Sanders

~~Topic:~~ Rash Decisions Advertising Communication-Flash Communication Server  
[[“MX”?]] aApp-wrap-up [[Chgs ok?]]

**Summary:**

Several “what else could we have done” enhancements were discussed, including a “rejected connection page,” more sophisticated streaming, email contact, and recording sessions.

**Ideas for ~~Improvement~~improvement or ~~Related-related Applications~~applications:**

If a user cannot connect because the session is full, instead of being frozen out, she/he could be sent to a page explaining that the host is currently using the application or that the conference is filled. This enhancement would be fairly easy using either client- or server-side scripts.

Currently, the audio and video streams are passed from the host to ~~the~~ all of the guest ~~guest~~-module users. However, only a single guest is passed to the host and none of the other guests [[Unclear—do you mean “only a single guest is passed to the host and not to any of the other guests”? Or if you mean simply “only a single guest is passed to the host,” just delete “and none of the other guests.”]]. (The app was set up this way to keep bw [[What’s this?]] down, but we might want to re-think this.) The text chat works fine with multiple users.

When a guest or host signs on, it would be nice if they could make use of an email module as a sort of reminder. The user could click an email button to let the others in the conference know she/he is there if no one has signed on yet. ~~Use~~ CF, PHP, or ASP.NET for a mailer.

Ability to record a session for later use or playback. How could we record a session into a single FLV file using more than a single stream? Could the streams some-how be filtered by the mic activation so that one stream or another is constantly appending an FLV file?

I file our post-mortem meeting notes. ~~Always b~~Before beginning a new project, we always get out the ideas/post-mortem files. Depending on the nature of the project, we ~~would~~ either find them useful or not, but generally, we always experiment with the ideas. Of course we all wonder why so many of our good ideas come when we are finishing a project and not when we begin one.