Tech Info Library

HyperCard: Sending Strings Using Apple Events (8/92)

Article Created: 28 April 1992
Article Reviewed/Updated: 28 August 1992

TOPIC ----------

I want to update an index over a network using identical HyperCard stacks. I use this handler:

```
on incoming data
  if data is not empty then put data into cd fld "index" of cd "topics"
end incoming
```

I send the contents of a large card to incoming by the following:

```
 send quote& cd fld "index" of cd "topics"&quote to remoteHC
```

where remote HC is the network address of the destination stack.

The script works fine in simple cases of "data". However, when sending the cd fld "index", I get an error that "Incoming could not understand QT", where QT is the first word of the second line of cd fld "index". The quotes on either end of the cd fld contents is not sufficient to distinguish the fld as a single entity; that is, everywhere I have a carriage return, the incoming handler thinks it's a separate argument and gets confused.

Sending each line of the cd fld "index" separately to avoid confusing the incoming handler works. Is there any way to send the contents of a field that has carriage returns as a single argument to a handler?

DISCUSSION ----------

The problem is really not related to Apple events. As an example, take the same script and field and try sending them to another stack without using Apple events (send "blah blah blah" to stack "blah"). You will find that returns can not be part of a string parameter when sent to another stack. Therefore the problem is caused by this constraint and not by an Apple event constraint.

Here are two recommendations for avoiding the problem.

First, create an "on AppleEvent" handler at the stack level of the receiving stack. With this created you simply send the whole field to your

receiving stack.

```
  on mouseUp
    send cd fld "data" to program otherApp
  end mouseUp
```

When the "AppleEvent" handler is activated in your receiving stack, you simply extract the contents of the sent field from the "data" parameter and put it into your location field.

```
on AppleEvent
  request AppleEvent data
  put it into cd fld "myField"
end AppleEvent
```

The only problem with this solution is that all Apple events will be trapped by this handler. If you are only planning on sending one Apple event or you are willing to parse out all potential Apple events in this handler, this solution is OK. If not you might want to try the next solution.

Using the same "incoming" handler you presently have, change the nature of the "data" parameter to represent the id (or number or name) of the field in the sending stack which holds the data. Then, in the "incoming" handler request the contents of this field from the sending stack. For example:

```
on incoming data
  request data from program sendingProgram
end incoming
```

The script you would use in the sending stack would look something like this:

```
on mouseUp
  answer program ""
  put it into otherApp
  send (incoming & quote & "cd fld id 1" & quote) to program otherApp¬
  without reply
end mouseUp
```

This article is adapted from the Claris Tech Info database.

Keywords: <None>

==========

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 14530