

QUICK RESPONSE FORM GENERATOR

INTRODUCTION

Life is full of simple questions:

What kind of food do you want to eat tonight?

Can you pick up Sarah from ballet lessons?

Does that red outfit make me look fat?

And many times they have simple answers. So basic are the question and answer that they can even be represented on a handheld device like a PDA or a cell phone. That's also desirable in that many times the only way to reach someone to ask them the question is via a mobile device.

The Quick Response Form Generator is a feature that gives Eudora users a fast and easy way to ask someone a question that can be answered by any Internet-enabled device. Since it can be used to communicate with cell phones it provides a reason for users to purchase/use cell phones, which is good for Qualcomm as a whole.

GUIDELINES

Here's a list of desired capabilities and restrictions for the QRFG:

- **Single question**
We want to limit the form to just one question to make responses easy.
- **Multiple-choice answers**
It's not very efficient to enter text on PDAs and especially on cell phones. It takes a while to enter even short text responses on cell phones (22444833#633).
- **Pre-defined templates for easy generation**
We want to have commonly used responses available at the click of a button: Yes/No; Yes/No/Maybe; ratings like 1-5; times like 11am, noon, 1pm and 5pm, 6pm, 7pm.
- **Customizable responses**
No matter how many templates we provide, there will always be a need for users to come up with canned answers on their own.
- **Access to previously composed forms**
Some questions are asked frequently with little or no change.
- **No summary/tallying ability now, but maybe later**
This means some sort of question ID.

USER INTERFACE

There's just one dialog for QRFG, but it's not trivial. The user can start generating a response form by selecting the Response Form menu item under Tools (Windows) or Window (Mac). Here's what the dialog looks like when the user first starts out:

The screenshot shows a 'Response Form' dialog box with the following elements:

- Previous:** A dropdown menu currently set to 'None'.
- To:** A text input field for email addresses.
- Question:** A text input field for the question text.
- Responses:** A dropdown menu currently set to 'Yes,No'.
- Response Grid:** A 3x3 grid of input fields. The first row contains 'Yes' and 'No' in the first two columns, with an empty field in the third. The remaining two rows are empty.
- Buttons:** 'Send' and 'Cancel' buttons at the bottom.

The **Previous:** combobox is where the user can retrieve previously composed forms. The name in the list will be the question followed by a comma-separated list of the responses in square brackets (e.g. “*Who is your favorite Stooge? [Moe,Larry,Curly,Shemp]*”). The default value is None. Selecting an item from the list will fill in the rest of fields using that previous form.

The **To:** field is where the user will type the email addresses of the intended respondents of the question. This field is similar in syntax to the To:, Cc:, and Bcc: fields in composition messages, allowing nicknames and using commas to separate addresses. It is also similar in that auto-completion will be enabled for this field so as to make it easier for users to enter commonly used addresses.

Question: seems pretty intuitive. It should do spelling and MoodWatch checking, given those options are turned on for normal composition. When the Question: field contains more than 80 characters, a line of static text will appear below it that says, “Mobile device users may find long questions to be unwieldy.”

As mentioned in the guidelines above, there will be pre-defined **Responses:**, which will be available in the combobox to the right. Here’s a starting list for now:

- Yes, No
- Yes, No, Maybe
- True, False
- 1, 2, 3, 4, 5
- Mostly agree, Somewhat agree, Neutral, Somewhat disagree, Mostly disagree
- Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

- 11am, 11:30am, Noon, 12:30pm, 1pm
- 5pm, 5:30pm, 6pm, 6:30pm, 7pm
- Loved it, It was okay, Shrug, I've seen worse, Hated it

No doubt there will be more response sets that should be added to the list. Implementers should also consider hidden options/resource strings that make this list easy to extend.

The default value is "Yes,No". The user can then add, remove, and edit responses in the edit controls below. What's shown here is an implementation that has a fixed number (9) of responses. Extra credit for the developer who adds/removes controls on-the-fly as the user types in/deletes text in the responses.

When the user is happy with what he's filled in, the last thing to do is click on the **Send** button. This will generate an outgoing message behind the scenes, and dismiss the dialog. What happens with that message depends on the Immediate Send setting: either sending right away or queuing in the Out mailbox for the next time that messages are sent. Holding down Shift while clicking on the button will do the opposite of the Immediate Send setting. If the message is queued, advanced users can go in to the Out mailbox and edit the message (e.g. change priority).

Also after clicking the **Send** button, the recipient list, question, and responses will be saved so that the form can be reused again via the **Previous:** list. We will save the last ten (maximum set by a hidden option) forms in the list.

RESPONSE FORM MESSAGE FORMAT

As mentioned in the above section, the result of the user finishing the form is that an outgoing message gets created. The main challenge is that the format must be something that can be understood and replied to easily by existing mail clients. HTML has the capabilities we need, and is widely deployed in mail clients.

We need some on-the-wire way to identify that this is a response form so devices that know about this format can take advantage of their input features (e.g. a cell phone may use a jog dial to cycle through the responses). For that reason, and for future tracking of responses, the Content-Type: header will get an extra parameter that notes that this is a response with an ID that identifies which response it is. The parameter will be "Response-Form" (case independent, like all good little MIME parameters are) and will take a value of a stream of alpha-numerics. It might be advantageous for the generating client to create IDs of 8-digit hex numbers so that the ID can be kept in a 32-bit value.

The Subject of that outgoing message will be the question. The To: header will contain the respondents (To: field) of the form. The first line of the body will be the question. After that, each possible response will be listed, one per line, as a mailto: hyperlink that will return that particular response. The order of the responses will be the same as they were entered in to the form. The format for the hyperlink is **mailto:ReturnAddressOfResponseCollector?Subject=QR:ID R:Response Q:Question**. The reason for the format of the subject is so that questioner can do some simple filtering

and sorting on the responses, and see the answers as a group without having to open the messages up.

Here's an example message:

```
To: Launcelot@RoundTable.uk, Robin@RunningAway.uk,
    Galahad@RoundTable.uk, Arthur@Camelot.uk
From: Bridgekeeper <keeper@bod.uk>
Subject: What...is your favorite color?
Content-Type: text/html; Response-Form=AD0482FC

<HTML><BODY>
What...is your favorite color?<BR>
<A
HREF=mailto:keeper@bod.uk?Subject=QR:AD0482FC%20R:Blue%20Q:What..
.is%20your%20favorite%20color%3F>Blue</A><BR>
<A
HREF=mailto:keeper@bod.uk?Subject=QR:AD0482FC%20R:Yellow%20Q:What
...is%20your%20favorite%20color%3F>Yellow</A><BR>
<A
HREF=mailto:keeper@bod.uk?Subject=QR:AD0482FC%20R:Blue,%20no,%20y
ellow!%20Q:What...is%20your%20favorite%20color%3F>Blue, no,
yellow!</A><BR>
</BODY></HTML>
```

There should be no signature assigned to the message, as cell phone users will frown upon any extra junk.

Allowing the user to edit the message can cause problems with the format. We will punt and say that users are free to screw up the form because editing is an advanced operation. One exception we may deem to handle is personality changes, because they may be common. For these forms, a personality change doesn't mean just a simple change of the From: header. The response HREFs must change as well. That could be achieved by saving a bit in the summary for outgoing messages that indicates whether this is a response form or not, and then going through all HREFs when the personality changes in a response form message. It may also be useful down the road (e.g. for gathering, tallying responses from the original) to be able to tell that an outgoing message is a response form.

We're not really sure what percentage of mail clients out there can handle HTML email messages, and of those who can't handle HTML it is unclear how many can do something reasonable with mailto: URLs (e.g. let the user click on them to create a new message). Given that the mailto: URLs are the bulk of the message, the conversion to plain text produces just about the same content, which is not very good for mail clients that can't handle URLs. We could generate a very stripped down version of the form, which did not include the mailto: URLs at all. It would just contain the question on the first line followed by the text of the responses on subsequent lines. However, that would mean that the format of the response would be variable, even for those mail clients who could handle mailto: URLs. Another part in a multipart/alternative, in addition to the text/plain and text/html parts, would be a nice way to go. However, some commonly

used mailers out there (Outlook Express and Communicator) wind up making an attachment out of unknown parts in multipart/alternative, which makes this undesirable.

It seems as though the best route to take is to allow the user's settings on sending plain/html/both to make the decision, and have the plain text part include the mailto: URLs so that format of the response is kept as uniform as possible. The only disadvantage to this is that respondents with mail clients that don't handle HTML or mailto: URLs will have to wade through all the formatting. Their responses were never going to be uniformly generated anyway. Whichever way is implemented, all parts should have the Response-Form parameter to the Content-Type: header.

FOR FUTURE CONSIDERATION

Here's a list of things that won't make the first cut, but that we may want to implement in a future version:

- **Personality selection**
As mentioned previously, the advanced user could open up the message from the Out mailbox and change the personality, although it would have to be done in several places. We could add a Personality combobox to the form.
- **Character styles of question and response**
Things like bold, italic, underline, color, font, while the form is being filled out. Again, another thing the advanced user can do it with the current design by opening up the message from the Out mailbox and applying the styles, but it might be nice to do it in the form itself.
- **Response summary**
The Response Wazoo! It could show all the questions that have been sent, the count of each individual response, and what response each person gave.
- **Content Concentrator**
There may be some ways to tie this in with the Content Concentrator feature in order to show all the responses in one message.
- **Instant Messaging**
What's described here is starting to overlap with the sending of IM messages. Maybe this becomes just a piece of a new way to communicate through other means besides email.
- **"Call me" response**
The cell phone-based clients (e.g. Bonsai) could add a default response to the end of all questions received that says something to the effect of, "Call me at (555) 123-4567 to discuss this further."