

Evaluator's Instructions and Tasks

To do ahead of time.

Review the Product Claims. An IBM research group is experimenting with interfaces that directly mimic real world products. One of the products is the **IBM RealPhone**, which they describe as follows:

Get Real with IBM RealPhone! Welcome to the future; one without distracting windows and menu bars. The RealPhone is an experiment in user interface design for a new, real-world user interface style. This preliminary design implements some elements of the style we are developing. Try it out, and tell us what you think.

Real World Appearance. You will not need to guess about what the RealPhone is supposed to do. If you can use a telephone, you can use this software. Novice users can use it immediately, but expert users can learn shortcut keys and other advanced ways of using the interface to make it more efficient. It is easy to learn and fast to use.

Part of the appeal of the RealPhone is its visual appearance. It is a fully-rendered, three-dimensional phone. When it is off the hook, the phone shows that. When it is dialing, the phone shows that too.

Because it is software, RealPhone can do more than a real telephone without breaking the visual metaphor. It has a built-in drawer for holding your most-used telephone numbers. The drawer works like a physical object, it slides in and out, but you can dial the numbers just by clicking on them. Try that with your paper organizer! Also, the body of the phone can be extended or shortened with a toggle switch. This allows you to have the phone on your desktop at all times to receive calls and extend it when you want to use the advanced dialing features.

No Window Controls. The most obvious missing pieces of this Windows™ app are the rectangular window border and controls. These items are included on nearly every existing Windows program. While they give you control over the "little boxes" (windows) on your computer screen, they are not really part of the task of writing a document, editing an image, editing a database, or sending a note. They also drive visual designers nuts, because they completely disrupt the visual metaphor of the application. The RealPhone is an attempt to produce a useful, robust application that breaks the confines of the little boxes and provides the user with an uncluttered view of the task at hand.

Install Real Phone on the computer. Make sure it works. Familiarize yourself with it.

Know how to Simulate the system (Wizard of Oz). The phone does not produce any auditory feedback, nor does it really dial any numbers. Your job as an experimenter is to mimic what the computer would actually do i.e., the dial tones, the person on the other side, etc. People will buy into this illusion, and will respond as long as you mimic the phone consistently and assertively. To do this properly, you will need to know how to respond to all situations! Note that this lack of sound/dialing is a limitation of the experimental setup, not with the RealPhone system.

Selecting Tasks. You should come up with *at least* six other reasonable tasks to give to subjects, preferably more. The best place to get good tasks for people to do is from the list of key tasks identified in the task-centered system design process. Tasks should also be selected to investigate different parts of the system functionality. Note that IBM makes some very strong claims in the description above: test these claims with your tasks! However, it is up to you to decide what set of typical tasks should be given to the users.

To get you going, a few sample tasks are included in the methods below. They are phrased as directions that will be given to the user. Note that we start with some very very basic tasks - if users have problems here, it is unlikely they will ever get to the advanced features of it. You should come up with *at least* six other reasonable tasks to give to

How to develop tasks. See the examples in the instructions. The experimenter should explore the system claims and the system itself. If the experimenter suspects problems, develop a task that exercises that part of the system. Some prime task areas are:

- Speed dials

- have them set up a speed dial for more than 10 people. Can they? How long does it take before people give up?
- have them add a phone-number only (or a name only) to the speed dial.
- can people set it up to handle long numbers (e.g., international calls), numbers with pauses e.g., to get an exterior line after dialing 9, or to wait before automatically entering a calling card number (note they will have to use help to solve this, but don't tell them that!)
- Window management
 - Have them get the window out of the way, e.g., iconify, shrink, etc.
 - There is a right menu option that will cause the real phone to always stay on top. See how useful this is.
 - Have them do some basic window operations: resizing, moving, expanding, shrinking to icon, closing etc. Note that these must be phrased in task terms
- Dialing
 - Can they figure out how things such as -, () and spaces handled?
- Open-ended tasks.
 - If time allows, ask people to set their own goals and to try to achieve them.

Create questionnaires. *Pre-test questionnaires* must at least probe for people's experience with the computer they are using, whether they have heard about or tried the real phone system previously, whether they have used software that allows them to make phone calls on their computers, whether they have previously used phones with speed dialing capabilities, etc. *During-test questionnaires* should probe for information and reaction about the task the person is trying to do while the task is still fresh in their mind. *Post-test questionnaires* should probe for reflective information, summary statements, and comments. These should include questions that ask people how satisfied they are with the system (e.g., "I would use RealPhone regularly if it was available on my home / office computer and if it was integrated with my phone system" (Strongly agree, agree, neither agree nor disagree, disagree, strongly disagree).

Instructions

Get the equipment set up ahead of time. The workstation, software, all forms, etc.

Bring the subject in.

Give the subject the consent form to read and sign.

Administer [Gomol's user observations guidelines](#) to subjects.

Give them the **pre-test questionnaire**

Conceptual model formation. Have them explain all the controls they can see on the Real Phone without interacting with it.

Give them each task, following each with the during-test questionnaire.

Task 1. Phoning home. Use the RealPhone to phone your house. Because RealPhone is not actually connected to the telephone, you will not hear any dial tones or phone numbers. Tell us what you think it would be doing as you do your actions.

Reason for choosing this task (not to be read to subjects). This is the most basic operation of phone use. It is frequent and important. Note that IBM claims "If you can use a telephone, you can use this software". Thus anyone should be able to do this first time around without hesitation. Did people need to look up help to do this basic task? If so, can they understand the help text?

Note to observers (not to be read to subjects): With real phone:

-you can pick up the handset, and then dial, or

-you can dial the number first, but you have to pick up the handset to actually call the number

Task 2. Speed Dial. Add your home number to the speed dial

Reason for choosing this task. This should also be very straightforward, as the phone is obviously set up for speed dialing. Things to notice: Can they add the number they had just dialed in task 1, or do they have to re-enter it? Is the interface immediately obvious on how to do it?

Task 3. Phoning an acronym. Phone IBM to tell them that you are using this system. You remember that

they advertise their number as 1-800-CALLIBM

Reason for choosing this task. This is a rare task, and probably not that important. However, its advertised as one of the features of RealPhone.

Note: With RealPhone you can enter letters instead of numbers e.g., G or H or I is the number 4. Note as well that, unlike most phones, there are no letters displayed on top of the number pad.

Other tasks you could develop. The experimenter should explore the system. If the experimenter suspects certain problems, the experimenter can develop a task that exercises that part of the system. Some prime areas for tasks are:

- Speed dials
 - have them set up a speed dial for more than 10 people. Can they? How long does it take before people give up?
 - have them add a phone-number only (or a name only) to the speed dial.
 - can people set it up to handle long numbers (e.g., international calls), numbers with pauses e.g., to get an exterior line after dialing 9, or to wait before automatically entering a calling card number (note they will have to use help to solve this, but don't tell them that!)
- Window management
 - Have them get the window out of the way, e.g., iconify, shrink, etc.
 - There is a right menu option that will cause the real phone to always stay on top. See how useful this is.
 - Have them do some basic window operations: resizing, moving, expanding, shrinking to icon, closing etc. Note that these must be phrased in task terms
- Dialing
 - Can they figure out how things such as -, () and spaces handled?
- Open-ended tasks.
 - If time allows at the end of a trial, you can ask people to set their own goals and to try to achieve them.

Give them the post-test questions

Interview them about anything interesting you saw that you want to discuss further.

Project Title: Real Phone - Evaluation of its Usability
Investigator(s): Saul Greenberg
Sponsoring Company: Saul Greenberg Consulting, 220 6087

This consent form should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, feel free to ask. Please take the time to read this carefully and to understand any accompanying information.

Description and conditions of the study.

IBM has developed a software product called "RealPhone" that is a software equivalent of a telephone. The purpose of this study is to discover what problems people may have when using it. You have been chosen to help us both because you have used Microsoft Windows, but you have not used the RealPhone product before.

You will be given the RealPhone software, and asked to perform a series of tasks with it. The entire process will take about one hour. As you perform the tasks, you will be asked to 'think aloud', where you say what you are thinking as you are doing it. You will also be asked to fill out a short series of questions before, during, and after the study. At the end, we will interview you about your experiences. During this, we will videotape what you are doing on the screen and capture your voice (although your face will not be captured on tape).

You may feel uncomfortable or awkward if you have trouble doing some of these tasks. However, we are testing the system, not you. If you have any problems with any of the tasks we ask you to do, that is exactly what we are looking for because we can then repair these problem areas and improve the product. Still, if you find this test objectionable in any way, you are free to quit at any time.

When discussing the results of this study, we will never refer to you by name or show your picture. In fact, we do not record your name on any of our records except on this sheet. Records and videotapes will be kept in a secure area and will only be disclosed to other members of the evaluation and development team. No other people will have access to this material.

Your participation is completely voluntary, without any payment.

Your signature on this form indicates that you have understood to your satisfaction your participation in the research project and that you agree to participate as a subject. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from the study at any time. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact: Saul Greenberg, 220 6087

Participant's Signature Date

Investigator Signature Date

Witness' Signature Date

A copy of this consent form has been given to you to keep for your records and reference.

-
1. I use a computer:
- a. almost every day
 - b. a few times a week
 - c. a few times a month
 - d. rarely, if ever
- other:*
2. When I use the computer, I usually use:
- | | | | |
|----------------------|--|---------------------------------------|---------------------------------|
| a. Microsoft Windows | <input type="checkbox"/> almost always | <input type="checkbox"/> occasionally | <input type="checkbox"/> rarely |
| b. A text editor | <input type="checkbox"/> almost always | <input type="checkbox"/> occasionally | <input type="checkbox"/> rarely |
| c. An email system | <input type="checkbox"/> almost always | <input type="checkbox"/> occasionally | <input type="checkbox"/> rarely |
| d. A web browser | <input type="checkbox"/> almost always | <input type="checkbox"/> occasionally | <input type="checkbox"/> rarely |
- e. *Other:*
3. I have regularly used on my computer: (check those that apply)
- a. a software-based fax machine
 - b. a software-based telephone dialing system
- c. *other:*
4. Regarding the RealPhone system: (check those that apply)
- a. I have never heard of it before
 - b. I have never tried it before
- c. *other:*

1. What did you like most about RealPhone?

2. What were your major annoyances / problems with RealPhone?

3. If we could only fix 3 things with RealPhone, what do you recommend we fix?

a.

b.

c.

Methods for running a basic observational study

Initial and developing conceptual model. Ask people to describe what they see on the screen and what they think easy system feature does. The reason you are doing this is to see what initial conceptual model people have of the system, based upon their prior experiences and their interpretation of the visuals on the screen. You are looking for places where the model is incorrect and incomplete. Start doing this as soon as they see the main screen or successive screens.

Task Observatons. Give subjects the tasks in sequence on the system (remember, tasks were prepared ahead of time!). You usually start with easy basic tasks, and then lead up to harder ones. Never give them all the tasks at once - this means you can 'pull the plug' early if people are having a difficult time. There are two primary observation methods:

Think Aloud Method. The single subject is asked to say what they are doing as they are doing it, and they should elaborate on any problems they are having. For example, here is what a subject may say:

"I'm going to try to do this task ... OK, this is probably the menu item I should select. Hmmm ... It's not doing anything, what's wrong? Oh, I see, I have to double click it..."

Note: Talking aloud is sometimes uncomfortable and unnatural for people to do. It may also interfere with the task the person is trying to accomplish.

Constructive Interaction. This method involves two Subjects working together on a new task, with the observer taking notes as before. The difference is that the natural communication between the two subjects will replace the unnatural talking aloud in condition 2. Also, the differences between subject's knowledge may lead to interesting questions, explorations, and answers between them. The best match of subjects is a semi-knowledgeable person matched with a fairly new user, with the later being in charge of interacting with the system. Thus you hear the new user asking questions, and the knowledgeable one explaining how to do things (sometimes incorrectly!).

The observer must take notes of the subject's behavior and key comments (professional usability people often use a video setup), especially where things appear to break down (e.g., errors, problems, etc.). While the observer is allowed to encourage the subject to talk freely (i.e. "What is it you are doing now? Why did you do that?) the observer should not interfere or help the subject in any way, no matter how tempting!

Caveat: If subjects get stuck. While the experimenter should not help the subject with the task, there are a few exceptions to this rule.

- If a subject has problems getting started, record the problems and give them a hint to get going. This is OK, because if they can't get started, they will not be able to do the tasks!
- If a subject cannot complete a particular task after a reasonable amount of time, tell them to stop and start them on the next task. Or, give them a hint if they cannot overcome some conceptual problem necessary to trying out other parts of the system. Again, record all problems.

Getting stuck is discouraging for subjects. Try to give them an early success experience, and remind them that they can quit at any time for any reason if they wish.

Questionnaires.

Never ask people questions if you will not use the answer. Asking irrelevant questions wastes people's time.

- A *pre-test questionnaire* asks about relevant subject demographics (e.g., experiences with the operating system, the actual product).
- A *during-test questionnaire* asks subjects about aspects of the system / task, usually after they complete a trial. e.g., "Was this task: 1-very easy ... to 5-very difficult".
- A *post-test questionnaire* asks subjects about the tasks/system as a whole. Its a good place to ask comparative questions as well.

Post-test interviews. The observer should then interview the subjects about their beliefs on how they performed, where errors were made, where the system helped them, where the system was weak, etc. As before, the observer should be taking detailed notes. Use the things you saw in the previous conditions to guide your interview. You can also use the filled-in questionnaire as a discussion tool (i.e. why did you answer this way?). A good method to direct the interview is to review the video tape with them and ask them what they were doing at that point in time (e.g., when a problem occurs). Seeing their acts grounds them in the reality of what they did.

Discussion of guidelines for user observation

Cite as: User Observation: Guidelines for Apple Developers, Kathleen Gomoll & Anne Nicol, January 1990

Introduction. User testing covers a wide range of activities designed to obtain information on the interactions between users and computers. Most user testing requires considerable expertise in research methods, as well as skill in using complex data collection tools. For example, user testing techniques include: interviews, focus groups, surveys, timed performance tests, keystroke protocols, and controlled laboratory experiments. Of the many user testing techniques available, user observation is one technique that can be used by anyone with a concern for including the user in the product development process.

User observation involves watching and listening carefully to users as they work with a product. Although it is possible to collect far more elaborate data, observing users is a quick way to obtain an objective view of a product.

When to observe users. User observation should be an integral part of the design process---from the initial concept to the product's release. Software design that includes user observation is an iterative process; user feedback provides the data for making design modifications. As Figure 1 demonstrates, this iterative process assumes that preliminary human interface designs should exist prior to the development of underlying code. Interface designs should be tested frequently to determine which design should be implemented. Then, as the code develops, the entire product should be tested and revised several times.

Preparing for a user observation

Set an objective. Before you do any testing, you should take time to figure out what you're testing and what you're not. In other words, determine an objective for your test that focuses on a specific aspect of the product. By limiting the scope of the test, you're more likely to get information that helps you solve a specific problem.

Design the tasks. Your test participant will work through one or more specific tasks. These tasks should be real tasks that you expect most users will do when they use your product. The entire user observation should not run over an hour, so you should design tasks that focus on the part of the product you're studying. For example, if you want to know whether your menus are useful, you could design a task that requires the participant to access the menus frequently. After you determine which tasks to use, write them out as short, simple instructions.

Important: Your instructions must be clear and complete, but they should not explain how to do things you're trying to test. For example, if you want to find out whether users can navigate through your program easily, don't give them instructions for navigation. Or, if you want to know whether your interface is self-explanatory, don't describe how it works. This concept is extremely important to remember. If you teach your participants about something you're trying to test, your data will not be useful.

Decide upon the use of videotape. Although you can observe users effectively without using special recording equipment, you may want to use videotape to capture the entire session. By videotaping the session, you collect an enormous amount of valuable information that you can review and analyze after the test is over. If video equipment is not available, a tape recorder can be helpful for recording what is said during the test.

Determine the setting. The ideal setting for user observation is a quiet, enclosed room with a desk, the appropriate hardware and software, a video camera, and two microphones (one for you and one for the participant). Of course, you may not have all these things available when you need to observe; therefore, you should try to approximate the ideal setting as closely as you can. If you have to conduct the observation in a regular office, ask the people around you to keep the noise level down during the observation. The key is to make the environment as interruption-free as possible. Get the participants out of their offices, away from phone calls and people who might drop by.

Find representative users. When looking for participants, try to find people who have the same experience level as the typical user for your product. Don't ask people you work with regularly to be participants because they are probably familiar with your product or your opinions about the product. Generally, you should look for people who are familiar with the hardware you use but are not familiar with your product.

You may want to ask pairs of people to work together on your tasks. You'll find that people working in pairs usually talk more than people working alone, and they also tend to discuss features of the product and explain things to each other.

10 steps for conducting a user observation.

The following instructions guide you through a simple user observation. Remember, this test is not designed as an experiment, so you will not get statistical results. You can, however, see where people have difficulty using your product, and you can use that information to improve it. These instructions are organized into steps. Under most of the steps, there is some explanatory text and a bulleted list. The bulleted list contains sample statements that you can read to the participant. (Feel free to modify the statements to suit your product and the situation.)

1. **Introduce yourself.**
2. **Describe the purpose of the observation (in general terms).** Set the participant at ease by stressing that you're trying to find problems in the product. For example, you could say:
 - You're helping us by trying out this product in its early stages.
 - We're looking for places where the product may be difficult to use.
 - If you have trouble with some of the tasks, it's the product's fault, not yours. Don't feel bad; that's exactly what we're looking for.
 - If we can locate the trouble spots, then we can go back and improve the product.
 - Remember, we're testing the product, not you.
3. **Tell the participant that it's okay to quit at any time.** Never leave this step out. Make sure you inform participants that they can quit at any time if they find themselves becoming uncomfortable. Participants shouldn't feel like they're locked into completing tasks. Say something like this:
 - Although I don't know of any reason for this to happen, if you should become uncomfortable or find this test objectionable in any way, you are free to quit at any time.
4. **Talk about the equipment in the room.** Explain the purpose of each piece of equipment (hardware, software, video camera, microphones, etc.) and how it is used in the test.
5. **Explain how to "think aloud."** Ask participants to think aloud during the observation, saying what comes to mind as they work. By listening to participants think and plan, you can examine their expectations for your product, as well as their intentions and their problem solving strategies. You'll find that listening to users as they work provides you with an enormous amount of useful information that you can get no other way.

Unfortunately, most people feel awkward or self-conscious about thinking aloud. Explain why you want participants to think aloud, and demonstrate how to do it. For example, you could say:

 - We have found that we get a great deal of information from these informal tests if we ask people to think aloud as they work through the exercises.
 - It may be a bit awkward at first, but it's really very easy once you get used to it.
 - All you do is speak your thoughts as you work.
 - If you forget to think aloud, I'll remind you to keep talking.
 - Would you like me to demonstrate?
6. **Explain that you cannot provide help.** It is very important that you allow participants to work with your product without any interference or extra help. This is the best way to see how people really interact with the product. For example, if you see a participant begin to have difficulty and you immediately provide an answer, you lose the most valuable information you can gain from user observation—where users have trouble, and how they figure out what to do.

Of course, there may be situations where you have to step in and provide assistance, but you should decide what those situations might be before you begin testing. For example, you may decide that you can allow someone to flounder for at least 3 minutes before you provide assistance. Or, you may decide that there is a distinct set of problems with which you can provide help.

As a rule of thumb, try not to give your test participants any more information than the true users of your product will have. Following are some things you can say to the participant:

- As you're working through the exercises, I won't be able to provide help or answer questions. This is because we want to create the most realistic situation possible.
- Even though I won't be able to answer your questions, please ask them anyway. It's very important that I capture all your questions and comments on tape.
- When you've finished all the exercises, I'll answer any questions you still have.

7. **Describe the tasks and introduce the product.** Explain what the participant should do and in what order. Give the participant written instructions for the tasks.

Important: If you need to demonstrate your product before the user observation begins, be sure you don't demonstrate something you're trying to test. (For example, if you want to know whether users can figure out how to use certain tools, don't show them how to use the tools before the test.)

8. **Ask if there are any questions** before you start; then begin the observation.

9. **Conclude the observation.** When the test is over:

- explain what you were trying to find out during the test.
- answer any remaining questions the participant may have.
- discuss any interesting behaviors you would like the participant to explain.

10. **Use the results.** As you observe, you see users doing things you never expect them to do. When you see participants making mistakes, your first instinct may be to blame the mistakes on the participant's inexperience or lack of intelligence. This is the wrong focus. The purpose of observing users is to see what parts of your product might be difficult or ineffective. Therefore, if you see a participant struggling or making mistakes, you should attribute the difficulties to faulty product design, not to the participant.

To get the most out of your test results, review all your data carefully and thoroughly (notes, the video tape or cassette tape, the tasks, etc.). Look for places where participants had trouble, and see if you can determine how your product could be changed to alleviate the problems. Look for patterns in the participants' behavior that might tell you whether the product was understood correctly.

It's a good idea to keep a record of what you found during the test. That way, you have documentation to support your design decisions and you can see trends in users' behavior. After you've examined the results and summarized the important findings, fix the problems you found and test the product again. By testing your product more than once, you can see how your changes affect users' performance.

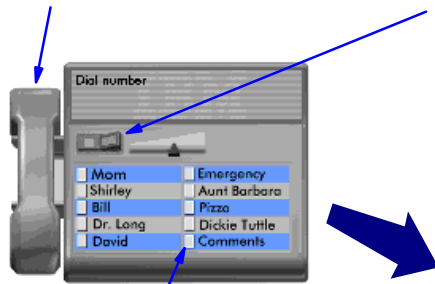
IBM RealPhone™

The easy to use telephone with the real-world style

The following diagram provides a quick overview of how to use all the IBM RealPhone features. Just scroll down further for a tutorial and helpful tips for using RealPhone.

Just click to pick up the receiver. It's as easy as that to answer an incoming call or to make a call yourself. Just click again to hang up!
(*'Enter'* to pick up, *'Delete'* to put down)

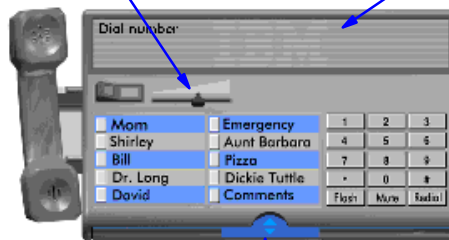
Just click to expand or minimize the function of your telephone...
(*'PgUp'* to expand, *'PgDn'* to minimize)



Just click to speed dial
(*'Alt'* + speed dial index)

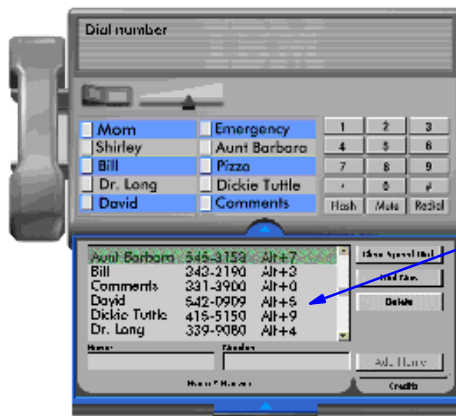
Slide the knob to set the volume
(*'Ctl+Right Arrow'* to raise volume, *'Ctl+Left Arrow'* to lower)

Dynamic display contains call information



Just click to slide out drawer for things like speed dial settings
(*'Alt+Enter'* to open and close)

Enter names and numbers here to set speed dials



* Virtual size may vary.



You can move it around, or even change its size when the pointer looks like this:

Tips for using IBM RealPhone with Windows 95™

The IBM RealPhone is an experiment in user interface design for a new, real-world user interface style we are currently developing. It is built using C++ and implements a reusable framework. Windows 95 limits the aspects of our design which can be demonstrated at this time. It begins to demonstrate the direction

we're heading in making user interfaces easier for everybody to use and easier for programmers to develop. Here are some tips for using the IBM RealPhone with Windows 95.

Define a shortcut key:

Use the properties page of the telephone's shortcut icon to assign a shortcut key. This will allow you to access the phone at any time. For example, the shortcut key will bring the telephone to the top when it is covered by other windows. Also, the phone will automatically answer if it's ringing when you press the shortcut key.

Using Microsoft Plus!™:

If you use Plus!, the telephone will be fully displayed during moving and sizing. Otherwise you will see the traditional outline box.

Special dialing characters:

You can pause during dialing by entering a comma (,) for a 2 second pause or semicolon (;) to wait for another dial tone.

Dial by character:

Telephone numbers made up of names or acronyms are easy to remember and difficult to dial on a real telephone (for example, 1-800-CALLIBM), but with IBM RealPhone you can just type the letters on the keyboard. RealPhone translates the letters to the desired phone number, and dials it for you!

Combining entries:

You can combine speed dial numbers with numbers you key yourself. For example, if 9 must be dialed to get an outside line, just enter 9 followed by the speed dial button.

"Closing" RealPhone in Windows 95:

In the future users will not need to "close" objects like RealPhone, but in Windows 95 this is still done. RealPhone supports this and other functions through the Windows 95 pop-up menu. Just click mouse button 2 over RealPhone to see the pop-up menu, and click button 1 over the option of your choice.

RealPhone is a trademark of IBM Corporation

Windows 95 and Microsoft Plus! are trademarks of Microsoft Corporation.

(c) copyright 1996, International Business Machines Corporation

Test-drive the new IBM RealPhone™!

What is RealPhone?

The IBM RealPhone was designed to work in much the same way as the telephone on your desk, wall, car, or pocket. But, unlike a physical telephone, it works through the computer. RealPhone takes advantage of the things computers are so good at - *remembering* people's names and numbers, *finding* them in a flash, and *entering* (dialing) them quickly - and lets you answer and place calls without picking up a physical handset.

You may notice some differences between the RealPhone and other telephony applications. For example, the RealPhone always looks like a telephone regardless of how you are currently using it. It doesn't need a window border around it; it sits right on your desktop. It's recognizable at any size, and you can make it as big or as small as you like without interfering with its functions. Even when it's as small as an icon, you can use it to answer calls or dial your speed-dial numbers, or to dial manually using the keyboard.

To get started...

The RealPhone is easy to use on your own, but the instructions below will help you get the most out of it by making sure you see all of its features. Before you start, install the RealPhone following the instructions provided with it. Then open the RealPhone by double-clicking on it. The phone will appear in its initial default state, the **full function view**.

- Call a friend.

You can dial numbers manually either by clicking on the phone's keypad with the mouse, or by using the computer's number keys. (You can even type in the letters for telephone numbers such as 1-800-CALLIBM.) When your friend answers, just start talking. The RealPhone uses your computer's microphone and speaker so that you can enjoy "hands-free" conversation. Before you wrap up your call, ask your friend to call you back in a few minutes.

- Disconnect.

When you've said goodbye, click on the RealPhone's handset to hang up.

- Receive an incoming call.

Your friend should be calling back about now. To answer the call, click with the mouse on the RealPhone's handset - you're connected.

- Speed-dial a number.
We'd like to hear from you. To leave us a message about using the RealPhone, press the last of the speed-dialing buttons (labelled "Comments"). Tell us what you think!
- Set a speed-dial number.
When you've finished your call (or even while you are talking) you can set speed-dial access to numbers you call often, or to emergency numbers:
 - Click on the handle of the drawer, which is located at the bottom edge of the RealPhone. The drawer will slide open, displaying an area where you can set the speed-dial buttons.
 - Type the name of the person to whom you want speed-dial access. Then fill in that person's telephone number in the **Number** field. (See the *Tips and Hints* document for instructions on entering numbers with pauses and waits.) Press the **Add Name** button. The name and number will appear in the list above the entry fields.
 - Press the **Set Speed Dial** key. The speed-dial buttons will flash. Click on the button you want to assign to this speed-dial number. The name and number you typed will appear beside its speed-dial button.

Close the drawer.

Click on the handle of the drawer to return it to its original position.

Change the appearance of the RealPhone.

The RealPhone can be displayed as it first appeared, with both the number pad and the speed-dial buttons, or in a simpler state, the **base function view**. To change between these presentations, click the toggle switch located just above the speed-dial buttons.

Stretch or shrink the RealPhone.

Move the mouse pointer over any edge of the phone. The pointer will change to a diagonally oriented *sizing pointer*. While this new pointer is showing, press down the left mouse button and drag the edge of the RealPhone to change its size. No matter how large or small you make it, the RealPhone can still be used.

After the test-drive

When you have finished exploring, please complete a copy of the survey you received with the RealPhone, and return the form to us as soon as possible. Thank you for participating. We look forward to hearing from you.

RealPhone is a trademark of IBM Corporation

(c) copyright 1996, International Business Machines Corporation



Interface Hall of Shame

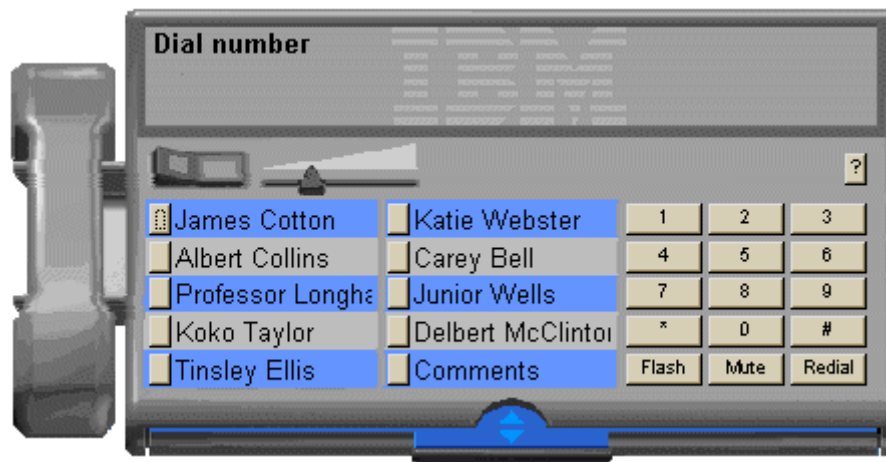
- IBM's RealPhone -

TM

Welcome to the future; one without distracting windows and menu bars. The RealPhone is an experiment in user interface design for a new, real-world user interface style...

So begins the IBM's literature on their experimental *RealPhone* application. Once we read this, we just knew we'd be creating a new chapter in the Interface Hall of Shame. Real-world metaphors have been attempted in the past, and have almost always met with failure. In this section, we take an in-depth look at the new interface, and attempt to illustrate why real-world metaphors rarely live up to their expectations.

Last updated 21-January-1997



The Hype

Welcome to the future; one without distracting windows and menu bars.

You will not need to guess about what the RealPhone is supposed to do.

Our User-Centric View

A Windows-based application without windows? Windows are used to separate tasks into manageable sections. You might have a word processor running in one window, and a spreadsheet application in another. A windows-based application without windows is analogous to a paper-based task without paper.

Similarly, we've found that menus are a valuable learning tool for new users. They organize the many functions of an application into distinct categories, thus serving as the primary means by which new users learn complex applications. If you remove the menus, you had better provide some other means of providing this valuable information. Unfortunately, *RealPhone* does not.

Put a telephone-type keypad on any application, and the user will pretty rapidly guess that it's a telephone application. Sure, having an image of a telephone handset helps, but it's not necessary. Make the handset a necessary control for the application, and you'll have a lot of users that are unaware that it's a control. Controls should *look* like controls, and they should appear manipulatable.

Here's where the metaphor starts to break down. No matter how similar

If you can use a telephone, you can use this software.

your program appears to look like a phone, it will always operate differently. When using a real phone, you pick up the phone, verify the dial tone, then dial your number. With *RealPhone* you dial your number, then point to the handset and *click* on it to start the call. Furthermore, to speed dial a number on a real phone, you pick up the handset, then press the speed dial number. On *RealPhone* however, you simply click on the speed dial number, which is likely to lead to a lot of inadvertent phone calls. Inadvertent mouse clicks don't happen when using real-world phones, but they occur frequently in computer-based applications.

Novice users can use it immediately...

Not likely. The application does not provide an area to type the number to be dialed. It displays numbers as they are typed, but because there is no control to receive the focus, there is no indication that you can type at all. Furthermore, while the interface provides command buttons for Redial and Flash, it does not provide a command button to initiate the call once the number has been entered. The user has to click on the handset, which is so non-intuitive that few users would ever consider trying it. In order to compensate for the non-intuitiveness of the interface, *RealPhone* relies on extremely lengthy tooltips to provide instructions. Many of the tips are so long they cannot be read in the display time for the tooltips (less than 3 seconds).

...expert users can learn shortcut keys and other advanced ways of using the interface to make it more efficient

Surprisingly, as part of their modernization of the interface, when IBM removed the control labels, they also removed essential interface components such as mnemonic access characters. While *RealPhone* does provide accelerator keys for the speed dial numbers, they are based on their numerical position in the list: to dial the first speed dial number, press ALT+1, for the second, press ALT+2, for the tenth, ALT+0. Unfortunately, the speed dial numbers do not have index numbers associated with them, so the user would have to count the speed dial positions to determine the numerical value to press.

Part of the appeal of the RealPhone is its visual appearance. It is a fully-rendered, three-dimensional phone.

Appeal is in the eye of the beholder. We would prefer that it look, and act, more like a computer application. In particular, we find *RealPhone's* lack of intelligence to be most **unappealing**.

RealPhone can do more than a real telephone without breaking the visual metaphor.

By modeling it after a real phone, IBM has unnecessarily **limited its functionality**. For example, the number of speed dial numbers on a real phone is limited by the size of the phone and economics. Virtual phones have no such limit. Why only 10 speed numbers? Why not 20, 50, or 100? They could simply have listed the names and let the user click on the desired name to call. IBM decided against utilizing the greatest asset of computers: unlike a real phone, computers are smart. A computer could very easily, for example, provide several *sets* of speed dial buttons that correspond to categories of numbers: "Clients", "Employees", "Friends", etc. Unfortunately, the real-world metaphor would not support such categorization, so IBM left it out of the application.

It has a built-in drawer for holding your most-used telephone numbers. The drawer works like a physical object, it slides in and out...

Maybe the phones at IBM have drawers underneath them, but not one of the phones we've ever used had a drawer. So much for the real-world metaphor. The resultant problem is that the user is provided no indication of how to add a name to the database, nor how to configure a speed dial button.

Also, the body of the phone can be extended or shortened with a toggle switch

Again, maybe the phones at IBM have toggle switches that modify the physical size of the phone, but we've never seen one. There is no real-world equivalent for this function. The toggle switch, like all controls in *RealPhone* are not labeled, nor is there a visual indication of its function. Only the curious would ever click on it.

The most obvious missing pieces of this Windows app are the rectangular window border and controls. These items are included on nearly every existing

The very components of windows that IBM eschews are extremely valuable controls and visual indicators. **That's why they are part of every existing windows program.** The border indicates that the window is resizable, and provides a means to resize it. The title bar provides a means to move the window, and when you have several programs running at once, provides the primary means of switching between

Windows program. While they give you control over the "little boxes" (windows) on your computer screen, they are not really part of the task of writing a document, editing an image, editing a database, or sending a note. They also drive visual designers nuts, because they completely disrupt the visual metaphor of the application.

applications. The "little boxes" are very important window management controls; they allow the user to quickly minimize or close the application to get to another. When you take away these elements, you take away valuable information, and very frequently used controls.

IBM's position on the "little boxes" is either pure arrogance or pure ignorance. The title bar standard exists to provide *consistent* access to these function across all applications. Suddenly, IBM concludes that this is a bad thing?

The basic problem with IBM's *RealPhone* is that IBM started out with the metaphor rather than the goal of optimizing the usefulness of the application. The most important aspect of a computer-based telephone application is **not** the ability to dial a phone number. You could simply provide a blank text box and a button labeled 'Dial' and have a more intuitive interface than *RealPhone*. The most important aspect of a telephone application is the ability to *manage* the names and numbers, part of which is retrieving a particular number quickly. IBM provided almost no administration functions in the application, not even the ability to edit an existing entry, nor the ability to search for a particular name. To do so would have "broken the visual metaphor". This is exactly the problem with applications based on real-world metaphors: they do not support functions that are not permissible in the real-world, but are not only feasible in computer applications, they are desirable. The task is to allow the user easy and rapid access to these information management functions. Unfortunately, *RealPhone* fails miserably in this regard.

Consider the process of assigning a name and telephone number to a speed dial position:

1. Open the 'drawer'
2. Type in name and number
3. Click on the Add Name button
4. Click on the Set Speed Dial button
5. Click on the desired Speed Dial button
6. Close the 'drawer'

Of course, this assumes the user somehow figured out this process, which can be difficult since *RealPhone* does not provide any instructions.

If you made a typographical error while entering the name or typed a name longer than 15 characters (the display limit on the speed dial labels), you'll have to:

1. Open the 'drawer'
2. Scroll to the name in the list
3. Click on the Delete button
4. Type in the corrected name and number
5. Click on the Add Name button
6. Click on the Set Speed Dial button
7. Click on the desired Speed Dial button
8. Close the 'drawer'

Be careful not to hit the Enter key when entering a new name and number. While anyone experienced with other Windows-based applications would expect the Enter key to be an equivalent to clicking the Add Name button, Enter will begin dialing whatever number is selected in the list of numbers, not the number being typed. If the user did not select a number in the list, *RealPhone* selects the first number in the list, even if the number is not visible. *RealPhone's* designers must have felt that default command buttons were simply another distraction in Windows applications.

If you hit the Delete button by mistake, you'll have to re-enter the information; there's no confirmation message to prevent inadvertent deletions, and there's no means to Undo an accidental delete.

Beyond the inappropriate application of the metaphor, *RealPhone* is replete with interface design problems, violating nearly every aspect of proper interface design. We found it so poorly designed that we almost decided against including it in the Hall of Shame. IBM's hype of the design can only lead to further disdain for the phrase "easy to use", and we feel, embarrasses the entire interface design profession.

IBM is currently offering an evaluation copy of *RealPhone* from their web site, at <http://www.ibm.com/ibm/hci/exhibits/3d/realphone.html>. Download a copy and judge for yourself.