



CHAPTER 1

Introduction

MOBILE Information Device Profile (MIDP) defines the Java application environment for mobile information devices (MIDs), such as mobile phones and personal digital assistants (PDAs). MIDP is part of the Java™ 2 Platform, Micro Edition (J2ME™). This guide offers advice for MIDP 2.0, which is specified in *MIDP 2.0 Specification* [19] from the Mobile Information Device Profile 2.0 (JSR-000118).

An application that runs in the MIDP environment is called a *MIDlet*. One or more MIDlets packaged together for distribution are called a *MIDlet suite*.

This book is for two groups of readers: those who are porting MIDP to a new device (*MIDP implementors*) and those who are creating MIDlet suites (*application developers*). Porting MIDP implementations to devices and designing MIDlets both have special challenges, which this book's guidelines address.

The guidelines are divided into three categories. The following paragraphs show the typographical conventions used for the guidelines, and explain the guidelines themselves.

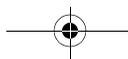


Strongly Recommend: Guidelines that, if not followed, could result in an unusable application.



Recommend: Guidelines that lead to an improved Java application in areas such as ease-of-use and portability. Disregarding this advice will not lead to an unusable application.

Consider: Guidelines that could lead to an improved Java application, but that do not necessarily apply to all applications and situations. Your circumstances may lead you to disregard this advice.





2 CONSUMER CHARACTERISTICS

The guidelines come from many sources, including usability studies, user-interface design principles, good coding practices, and experience implementing the MIDP specification and MIDlets.

This chapter introduces design considerations for MIDlets in the consumer market. Designing MIDlets differs significantly from designing desktop computer software. Some important differences include product domain, resource limitations, and the need to focus on ease of use.

1.1 Consumer Characteristics

Consumer-application users have different characteristics than users of desktop systems. Consumers are familiar with appliances that typically have simple, predictable UIs, such as phones, microwave ovens, and remote controls, and might feel uncomfortable dealing with anything they consider too “high-tech.” They expect consumer products to be predictable, easy to learn, and easy to use.

While electronic appliances are common today, widespread acceptance of new products is still hard fought. Eight of ten consumer products fail in the marketplace, often because consumers find them too difficult to use.

Application Developers and MIDP Implementors



Recommend: Make new devices and applications as familiar, responsive, easy to learn, and easy to use as possible.

1.2 The Nature of Consumer Products

It is important to consider the limited resources and particular input and output (I/O) mechanisms of a consumer device because they typically have less memory, smaller or lower-resolution displays, fewer colors, and different I/O mechanisms than personal computers. Software and hardware must be tightly integrated.

MIDP Implementors

Consider: Integrate your MIDP implementation with the device’s user experience. For example, coordinate the colors on the display with the colors of the physical components (such as the plastics), and the shape of





the buttons on the screen with the shape of the physical buttons on the device. Strive to make using a MIDP application seem like using a *native application* (a resident application already installed on the device).

1.3 Consumer-Product Domains

Specialization for particular activities is a strength of consumer products such as televisions and mobile phones. Even products that might be used for multiple activities have a fundamental purpose. For example, a mobile phone is primarily a communications device, even though it may also include a few games. Devices that are compelling and successful in the marketplace have the right focus and features.

The focus of a device (the tasks it performs and the setting in which it is used) identifies its *domain*. For example, the domain of a basic pager is information access. Its tasks are receiving and displaying phone numbers or short text messages. It is used in short sessions and can be carried to any location. The consumer must be able to retrieve information (a phone number or a short message) quickly.

A product's domain has implications for the UI design and the user experience. There are many domains for consumer products. This section covers three that are relevant to MIDP.

1.3.1 Information Access and Communication

Mobile phones typify consumer products in the information access and communication domain. For example, consider a consumer using a web-enabled phone to find a particular restaurant, call it, and make reservations. Figure 1.1 shows how a consumer might interact with an application to perform this task.

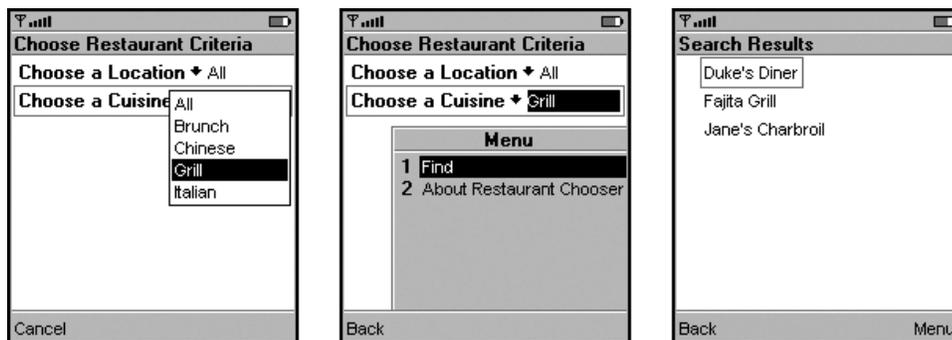


Figure 1.1 Using a Web-Enabled Phone to Make Reservations





4 CONSUMER-PRODUCT DOMAINS

The information access and communication domain has the following characteristics.

- Products are used for:
 - Short periods at a time (a few seconds to five minutes)
 - Specific tasks
- Tasks:
 - Are usually structured and directed
 - Might be interrupted, but the interruptions are likely to be task relevant, such as getting a call while looking up a phone number
- Consumers:
 - Are likely to concentrate during their interactions
 - Approach the product with specific tasks or goals in mind
 - Are motivated to complete tasks and want quick, efficient, easy experiences
 - May be skeptical of the device's network connectivity
 - May doubt the device's ability to communicate properly at all times

The characteristics of this domain lead to the following design considerations.

Application Developers

Consider: Create UIs that provide efficient experiences. Task completion is important and time may be a critical factor. Consumers must be able to complete tasks quickly and efficiently.



Recommend: Make screens work together. For example, an application should not return information to the consumer in one screen only to have the consumer enter it again a few screens later. It is inefficient and frustrating for a consumer to enter information that the application should already have.



Recommend: Let the consumer know the outcome of an interaction that takes place across a network. It is obvious when some interactions have completed successfully. For example, when a browser requests a web page, the information exchange has obviously completed when the page



is displayed. For other interactions, such as sending data to a remote location, the outcome might not be as obvious. Let consumers know that such data exchanges are progressing and when they are completed.

1.3.2 Business Functions

Products in the business functions domain help workers or operators perform very specific, job-related tasks. Examples include:

- Devices to help waiters take orders
- Applications to help assembly-line supervisors debug problems
- Devices to help delivery personnel plan routes and get customer signatures

These kinds of products can be thought of as vertical-market devices and applications because they target a specific task in a narrow field. Figure 1.2 shows an example application in this domain.



Grocery List:	
Eggs, 2 dozen	3.50
Milk, 3 half-gallons	6.00
Zucchini, 1 lb.	1.29
Tofu, 1 lb.	1.50
Spaghetti, 1 lb.	2.25
Rice, 5 lb.	5.25
Total	\$19.79
Signature:	
<input type="text"/>	

Figure 1.2 Using a PDA for Customer Signature Capture

The business functions domain has the following characteristics:

- Products are used for job-related tasks, for which training costs can be an issue
- Tasks are such that interruptions are possible but are likely to be task relevant
- Tasks might be completed incrementally
- Consumers usually have domain knowledge specific to the application

The characteristics and design implications of the business functions domain are similar to those of the information access and communication domain; however, applications in this domain need to be efficient and predictable. (See “Information Access and Communication” on page 3.)



6 CONSUMER-PRODUCT DOMAINS

1.3.3 Entertainment

Playing a game on a mobile phone while standing in line at the grocery store exemplifies an interaction in the entertainment domain; for example, see Figure 1.3.

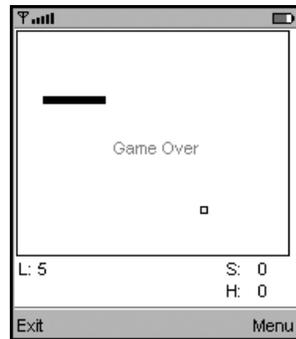


Figure 1.3 Using a Mobile Phone to Play a Game

Consumer products used for entertainment, such as games on PDAs or gaming devices, share these characteristics:

- Products are used:
 - In comfortable, low-stress environments
 - For 10 to 30 minutes or more at a time
 - For less structured tasks
- Tasks are such that interruptions are possible
- Consumers:
 - Are likely to have varying levels of concentration and attention during their interactions
 - Approach the products with a relaxed attitude
 - Are more interested in a pleasant experience than in performing a specific task





Design considerations for the entertainment domain include the following.

Application Developers and MIDP Implementors



Recommend: Create UIs that interfere as little as possible with the content. For example, an application's controls should take up as little screen real estate as possible so that consumers can see more of the screen. Consumers will have various levels of concentration. Forcing consumers to concentrate on the UI will make their experience with the product less enjoyable.

Application Developers

Consider: Create entertainment UIs that provide a pleasant experience in preference to an efficient one. Task completion time in this domain is not a critical factor because the consumer is not under significant time pressure.

Efficiency is a positive attribute, but in the entertainment domain, a pleasant and compelling experience is as important as efficiency. Predictability, however, is more important than efficiency.

Consider: Devote as much I/O bandwidth as possible to seeing, hearing, and interacting with the entertainment. The interaction of the consumer and the game is the product's most important feature.

1.4 Design Considerations for Consumer Products

Some principles for designing consumer-product UIs are independent of the product's domain. This section covers some of these principles. See *Information Appliances and Beyond: Interaction Design for Consumer Products* [1] for more information.

1.4.1 Simplify

Simplifying UIs is an objective for all designers. For consumer products, you can make a trade-off of functionality and choice against simplicity.





1.4.2 Functionality Versus Simplicity

The more functionality included in a product, the more difficult it is to learn and to use. On the other hand, gratuitous elimination of functionality can lead to a product that is too limited to support consumers' needs. Strive for the *functionality threshold*—the right collection and number of features to strike a balance between functionality and simplicity.

Application Developers and MIDP Implementors



Recommend: Use the 80/20 rule: Identify the 20 percent of the functionality that will meet 80 percent of the consumers' needs and optimize your design accordingly. After you have supported 80 percent of the consumers' needs, you can decide what, if any, of the remaining functionality to include based on other criteria (such as competitive edge and price). Do not change your design to provide the same level of access to these functions as to the critical functionality.

1.4.3 Choice Versus Simplicity

Choice is useful when it is appropriate, but too much choice can create a complicated, perhaps overwhelming, situation. For example, consider replying to an email. An email application could give you the choice of replying to the sender and including the original mail in the reply, replying to the sender without including the original mail, replying to everyone who received the original mail and including the original mail in the reply, and replying to everyone who received the original mail without including it in the reply.

Application Developers and MIDP Implementors

Consider: Reduce choice by providing reasonable defaults and removing nonessential options. For example, MIDlet suites can contain one or more MIDlets. A MIDlet suite containing multiple MIDlets forces the consumer to choose which MIDlet to run. If a MIDlet suite contains a single MIDlet, a MIDlet implementation could simply launch the suite, relieving the consumer of having to “choose” from a single element list. A MIDlet implementation that removes that choice point has a simpler design and meets consumers needs.





1.4.4 Make It Predictable

Usability testing has shown that consumers are much happier with systems that are predictable, even at the cost of efficiency.

Application Developers and MIDP Implementors



Strongly Recommend: Design your MIDP implementation or MIDlet so that consumers can predict what will happen when they take an action. Consumers rarely read documentation.

For example, assume that an address book application and an email application on a mobile phone have menus that display the available operations, and the address book has a menu element called New Entry for adding a record. The question for the email application is whether its list element for writing a new message should be called Write or New Message. Using New Message is parallel to New Entry in the address book, which makes it an attractive option. It makes the address book and email applications more consistent. Consumers, though, do not think about creating a new message; they think of writing to someone. For them, Write would probably be a better predictor of the command's behavior.



Recommend: If you must choose between predictable and efficient, choose predictable. Predictability is a better investment than efficiency.

MIDP Implementors



Recommend: Publish the runtime behavior of your MIDP implementation. This will help developers understand the conventions and behavior of your device so that they are better able to make their MIDlets predictable to users of your device.





1.4.5 Streamline Important Tasks for Efficiency

Application Developers and MIDP Implementors



Recommend: Minimize the amount of navigation and user interaction required to complete frequent or crucial tasks. These operations should require as little overhead as possible.

When you consider this recommendation remember that predictability is more important than efficiency in a consumer product.

One way to minimize navigation is to design applications with shallow hierarchies so that consumers do not find themselves going from screen to screen to screen, and then back, back, back.

Application Developers

Consider: If an application requires a sequence of steps from which consumers may need to retreat, enable them to go back multiple steps at a time in addition to returning to the previous screen.

1.4.6 Make It Responsive

Consumers expect immediate response to their input. When a response is not immediate, they can become annoyed, repeatedly press buttons, or assume the device is broken. They may stop using the device.

Application Developers and MIDP Implementors



Strongly Recommend: Have the device respond so that consumers are confident that each of their actions was received. For example, the device could confirm button presses with a click sound and in some circumstances an application might use visual feedback to tell consumers the system got their input.



Strongly Recommend: Avoid blank screens. Consumers think that a product is faster if things keep happening on the screen, as opposed to not showing anything until an operation is done.





Strongly Recommend: Minimize delays and if there is a delay give appropriate feedback. For example, display an animated indicator when downloading information, preferably one that shows how the operation is progressing. Providing dynamic feedback to consumers during delays keeps them engaged.

Responsiveness starts with immediate feedback and goes beyond it to intelligent responses.

MIDP Implementors

Consider: Have the product respond to common activities in a way that makes life easier for the consumer. For example, launching an application is a common activity. If a MIDlet suite contains a single MIDlet, it makes the consumer's life easier if the device automatically runs the MIDlet when the consumer launches the MIDlet suite (as mentioned previously).

1.4.7 Provide Constant, Unobtrusive Feedback

The users of consumer products need reassurance that the device is functioning correctly, even when nothing important is going on.

MIDP Implementors



Strongly Recommend: Ensure that something on the screen always shows that the product is on. The indicator should not be annoying or distracting. It could be something like a connection graphic or a power light.

For example, most mobile phones have a signal strength indicator, which is active when the phone is on. Its feedback is crucial when making phone calls. The indicator is positioned so that it does not interfere with the display of other information. Figure 1.4 shows the signal strength indicator in the MIDP Reference Implementation.



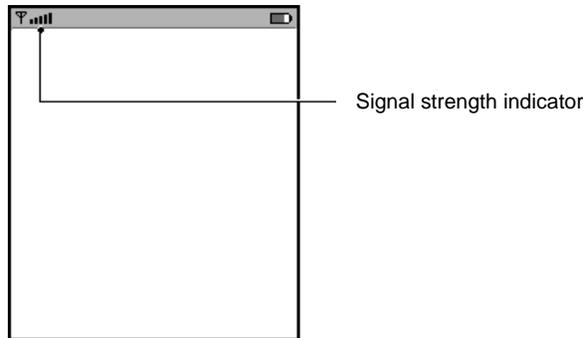


Figure 1.4 Signal Strength Feedback on a Mobile Phone

Application Developers and MIDP Implementors

Consider: Animation and sound can convey feedback to the consumer. In some cases, a slow animation might be more unobtrusive than other types of feedback. For example, an animation indicating that a product is connected to a network does not have to change quickly; it is more effective at a slow, even pace.

Figure 1.5 shows the images for a slow, even animation that indicates network activity.

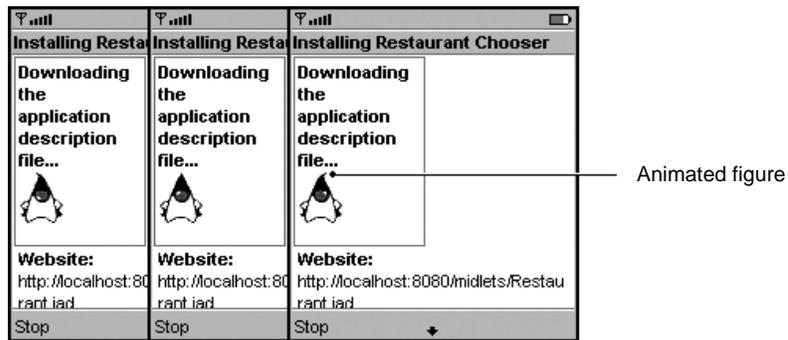
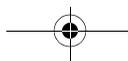


Figure 1.5 Animation Indicating Activity



1.4.8 Make Everything Interruptible

Consumers should *always* be able to cancel, power off, or simply interrupt the device in order to do something else, even when there is a message that requires a response on the screen or they are in the middle of a task. If a person is in the middle of an activity, the consumer should be able to start over at the beginning of that activity without penalty.

Application Developers and MIDP Implementors



Strongly Recommend: Enable consumers to interrupt an application no matter what state it is in.

For example, when a PDA shows a *modal alert* (a message on a screen that the user must dismiss), the consumer should be able to dismiss it not only by tapping a particular button but also by tapping outside the alert. Similarly, if a consumer is writing an email message and wants to go to another application, the device should automatically save the message as a draft without displaying error messages or prompts.

Allowing a user to interrupt anything keeps control of the product in the hands of the consumer. People are more comfortable with products they control.

1.4.9 Minimize Interruptions From MIDP and MIDlets

Consumers should not be interrupted and required to respond unless it is absolutely necessary. Confirmation messages, unnecessary feedback, and error messages that require a response detract from the user experience.

Application Developers and MIDP Implementors

Consider: Restrict interruptions to critical information.

Consider: Use less-obtrusive mechanisms, such as a *timed alert*, for providing feedback that is not critical. A timed alert is a screen with a message for the user; it is visible briefly and does not require a response. For example, if the consumer is in a Preferences screen and presses Save, a timed alert could be used to confirm that the preferences were saved.



It can be difficult to decide when a particular piece of information is critical. For example, consider consumers downloading applications onto mobile phones. If network connections are lost, the losses can be considered critical information that lets consumers know that they should reconnect. Another approach, though, is to have the devices automatically try to reconnect and unobtrusively display that status. The second approach keeps interruptions to a minimum. When deciding how to handle this situation, consider the cost to the consumer for the connection, as well as the consumer's time.

Application Developers and MIDP Implementors, Continued



Strongly Recommend: When using modal alerts to give consumers critical information, do not make the device appear to be broken. Use plain language to explain the problem. Test your error messages with real consumers.

This approach allows consumers to decipher critical information and maintains their confidence in the device.

1.4.10 Check Your Designs with Others

In any design process, it is valuable to periodically check your designs against the wants and expectations of your audience. You could do this by comparing your design to product-marketing requirements, visiting prospective customers, and evaluating competing products.

Application Developers and MIDP Implementors

Consider: Involving consumers in exploratory studies, blank model studies, and surveys are some of the best ways to check how well consumers like your product design. (Blank model studies allow consumers to design a product from scratch, without being given a design to start with. They remove a designer's preconceptions from the consumers' view.)

See "Bibliography" on page 243 for sources of information on these techniques.

