

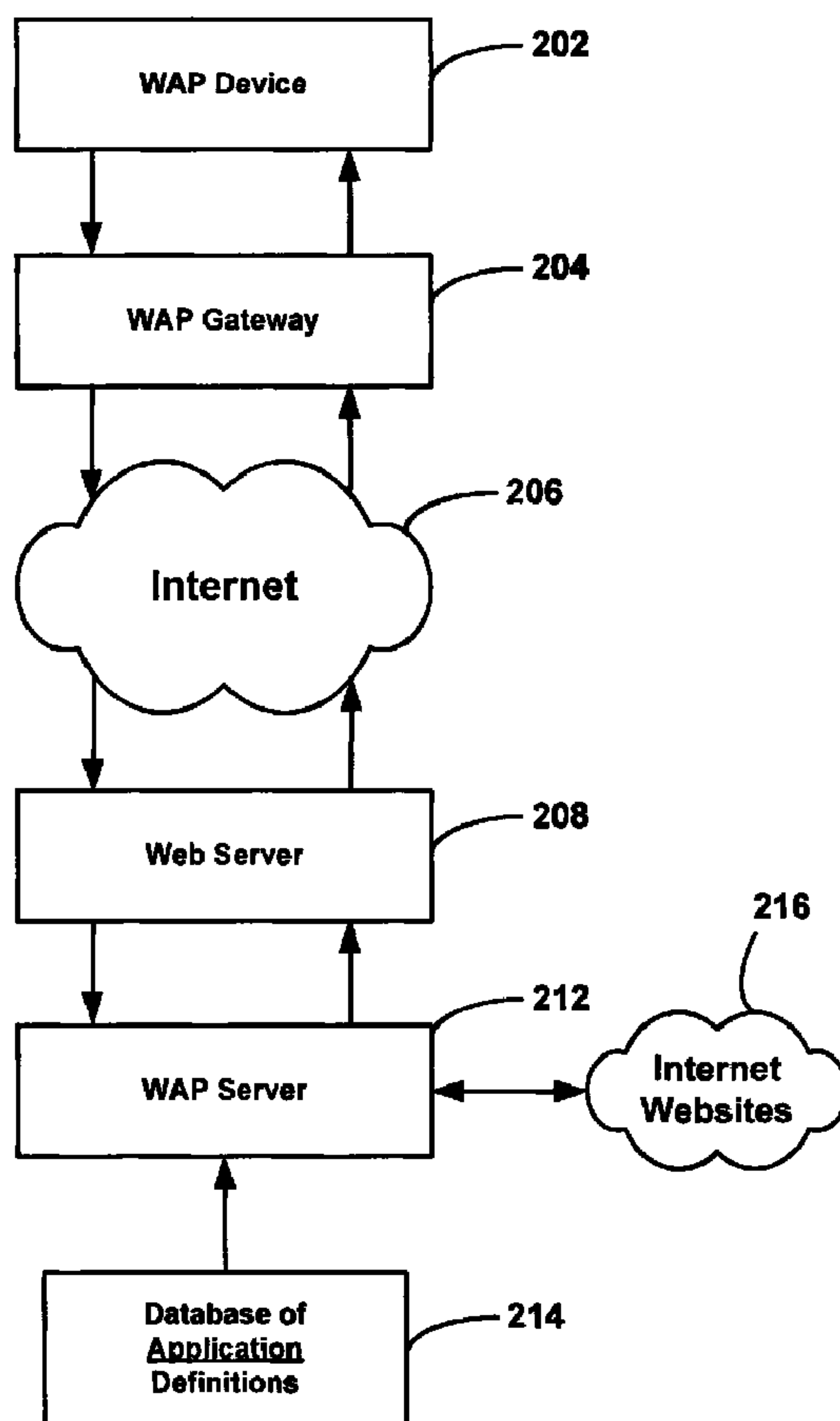


(22) Date de dépôt/Filing Date: 2001/08/14
(41) Mise à la disp. pub./Open to Public Insp.: 2003/02/14
(45) Date de délivrance/Issue Date: 2008/01/08

(51) Cl.Int./Int.Cl. *H04Q 7/20* (2006.01),
H04Q 7/22 (2006.01)
(72) Inventeurs/Inventors:
HO, RAYMOND, CA;
FUNG, EDWARD, CA
(73) Propriétaire/Owner:
OPTIMICRO TECHNOLOGIES, INC., CA
(74) Agent: SIM & MCBURNEY

(54) Titre : SYSTEME ET METHODE DE DEFINITION, DE CREATION ET DE DEPLOIEMENT D'APPLICATIONS SANS
FIL

(54) Title: SYSTEM AND METHOD FOR DEFINING, CREATING AND DEPLOYING WIRELESS APPLICATIONS



(57) Abrégé/Abstract:

A system and method for conveying a WAP application to a WAP compliant wireless device comprises a WAP compliant wireless device generating a WAP request. A WAP gateway generates a web request in response to the WAP request and conveys the

(57) **Abrégé(suite)/Abstract(continued):**

web request to a web server over the Internet. A WAP server reads application definitions from a database that are appropriate for generating a WAP response in response to the WAP request and dynamically generates WML code forming the WAP response using the application definitions read from the database. The WAP server transmits the WAP response to the WAP compliant wireless device via the web server, Internet and WAP gateway. A WAP application builder and method are also provided.

ABSTRACT

A system and method for conveying a WAP application to a WAP compliant wireless device comprises a WAP compliant wireless device generating a WAP request. A WAP gateway generates a web request in response to the WAP request and conveys the web request to a web server over the Internet. A WAP server reads application definitions from a database that are appropriate for generating a WAP response in response to the WAP request and dynamically generates WML code forming the WAP response using the application definitions read from the database. The WAP server transmits the WAP response to the WAP compliant wireless device via the web server, Internet and WAP gateway. A WAP application builder and method are also provided.

**SYSTEM AND METHOD FOR DEFINING, CREATING AND DEPLOYING
WIRELESS APPLICATIONS**

Field of Invention

The present invention relates generally to wireless devices and in particular to a system and method for defining, creating and deploying applications
5 for wireless devices.

Background of the Invention

Wireless devices such as mobile or cellular telephones, pagers and Personal Digital Assistants (PDAs) have gained wide acceptance in today's society.
10 Over the past years wireless devices of this nature have provided users with increased features including Internet access. Unfortunately, these wireless devices suffer limitations that inhibit the wireless devices from taking full advantage of Internet services. These limitations include low bandwidth, high latency, and connection availability of wireless networks as well as limited memory, execution speed, small
15 display screens and small keypads.

To standardize competing protocols, extend Internet services to wireless devices, and deal with some of the above-mentioned limitations, the Wireless Application Protocol (WAP) was established by the WAP Forum consortium. WAP compliant wireless devices typically execute WAP applications written in Wireless
20 Markup Language (WML). WML is similar to Hyper Text Markup Language (HTML), the language used to write most Internet applications executed by Internet browsers. Similar to HTML, WML has the advantage of being platform independent, provided the WAP compliant wireless device runs a browser program that can interpret the language properly.

25 When a WAP compliant wireless device initiates an Internet session and generates web requests, WAP applications are delivered to the WAP compliant wireless device in the form of decks of WML "cards". Each WML card is the functional equivalent of a web page. When a deck of WML cards is received by the WAP compliant wireless device, the WML cards are executed by the browser running
30 on the WAP compliant wireless device and displayed on its screen. Using the browser, a user is able to page through the deck of WML cards and generate additional web requests.

A number of different WAP applications have been developed since the inception of the WAP specifications, but unfortunately most WAP applications are still developed in a specific and ad-hoc manner. As a result, WAP application developers still encounter a number of unresolved issues.

5 As mentioned previously, WAP compliant wireless devices suffer from a number of limitations. In addition, there are substantial differences in the limitations of various WAP compliant wireless devices. As a result WAP application developers must keep these limitations in mind when developing WAP applications to be deployed across a variety of WAP compliant wireless devices. Inevitably, WAP
10 application developers take the lowest common denominator approach and write WAP applications for WAP compliant wireless devices that suffer the most severe limitations, knowing that if this is done, the WAP applications will be able to run on virtually all WAP compliant wireless devices. The end result is that most, if not all WAP applications, fail to take advantage the available resources of many WAP
15 compliant wireless devices.

Although the adoption of WAP standards was intended to make WAP applications platform independent, it has been found that many WAP applications behave differently when executed by browsers running on different types of WAP compliant wireless devices. Unfortunately, WAP applications provide little, if any
20 facility to change the WML code on-the-fly to handle particularities of various WAP compliant wireless devices.

During WAP application development, most WAP application developers use some form of editor to write the WML code. As a result, WAP application developers must learn and understand the syntax and semantic of WML
25 before they are able to effectively develop WAP applications. In addition, since WAP supports multilingual user interfaces using Unicode (an encoding standard for International languages), WAP application developers sometimes are required to deal with the issues relating to the interchange of Unicode with other encoding formats that exist in current systems. Furthermore WML does not support database access. It
30 is up to WAP application developers to use the necessary protocols to permit WAP applications to access database systems. The handling of database transactions and

error conditions therefore has to be dealt with each and every time a new WAP application is developed.

Unfortunately, to-date editors used to write WML code provide no means to expedite or facilitate WAP application development. Also, these editors
5 provide little, if any ability to interchange Unicode with other encoding formats. Furthermore, these editors provide little, if any means to shield WAP application developers from technical protocol details when developing WAP applications that require database access. As will be appreciated, a need exists to assist WAP application developers.

10 It is therefore an object of the present invention to provide a novel system and method for defining, creating and deploying applications for wireless devices.

Summary of the Invention

15 According to one aspect of the present invention there is provided a system for creating and conveying a wireless application to a wireless device comprising:

a wireless device generating a wireless application request;
a gateway generating a web request in response to the wireless
20 application request and conveying the web request to a web server over the Internet;
a server reading application definitions from a database that are appropriate for generating a wireless application response, in response to said wireless application request and dynamically generating code forming said wireless application response using the application definitions read from said database, said
25 server transmitting said wireless application response to said wireless device via said web server, Internet and gateway; and

a wireless application builder creating wireless applications in response to user input and storing the application definitions making up the logic and interfaces

of each built wireless application in said database, during building of a wireless application, said wireless application builder providing a visual wireless application creation platform thereby to enable said user to define and view the application definitions making up the wireless application being built, in graphical logic flow
5 form.

According to another aspect of the present invention there is provided a wireless application builder for creating a wireless application comprising:

a graphical user interface providing a visual wireless application platform enabling a user to define and view application definitions making up a
10 wireless application being built in graphical logic flow form; and

a wireless application generator creating application definition data defining the logic and interfaces of the wireless application being built in response to defined application definitions, wherein said graphical user interface presents windows for said application definitions.

15 According to another aspect of the present invention there is provided a method for building a wireless application comprising the steps of:

presenting a graphical user interface defining a visual wireless application creation platform;

20 defining wireless application definitions corresponding to a graphical logic flow diagram modeling said wireless application in response to user input;

presenting said application definitions in windows; and

creating application definition data in response to defined application definitions, said application definition data defining the logic and interfaces of the wireless application being built.

25 According to yet another aspect of the present invention there is provided a method of conveying a wireless application protocol (WAP) application to a WAP compliant wireless device comprising the steps of:

generating a WAP request using a WAP compliant wireless device;

conveying the WAP request to a WAP gateway;

-4a-

at the WAP gateway generating a web request in response to the WAP request and conveying the web request to a web server over the Internet;

at the web server generating a WAP request in response to the web request and conveying the WAP request to a WAP server;

5 at said WAP server reading application definitions from a database that are appropriate for generating a WAP response in response to said WAP request and dynamically generating wireless markup language (WML) code forming said WAP response using the application definitions read from said database, said application definitions comprising display, menu selection, option selection, user input, retrieve
10 data, insert data, update data and delete definitions; and

transmitting said WAP response to said WAP compliant wireless device via said web server, Internet and WAP gateway wherein said display definition includes instructions for generating WML code to display static text and WAP bitmap images on said WAP compliant wireless device; said menus selection definition
15 includes instructions for generating WML code to display a list of menu items as hyperlinks on said WAP compliant wireless device; said option selection definition includes instructions for generating WML code to display option items on said WAP compliant wireless device; and said user input definition includes instructions for generating WML code to obtain user input from said WAP compliant wireless device
20 and wherein said retrieve data definition includes instructions for generating WML code to retrieve data from a data source; said insert data definition includes instructions for generating WML code to insert data to a data source; said update data definition includes instructions for generating WML code to update data in a data source; and said delete data definition includes instructions for generating WML code
25 to delete data from a data source.

According to still yet another aspect of the present invention there is provided a system for conveying a wireless application protocol (WAP) application to a WAP compliant wireless device comprising:

-4b-

a WAP compliant wireless device generating a WAP request;

a WAP gateway generating a web request in response to the WAP request and conveying the web request to a web server over the Internet; and

a WAP server reading application definitions from a database that are
5 appropriate for generating a WAP response in response to said WAP request and
dynamically generating wireless markup language (WML) code forming said WAP
response using the application definitions read from said database, said application
definitions comprising display, menu selection, option selection, user input, retrieve
data, insert data, update data and delete definitions, said WAP server transmitting said
10 WAP response to said WAP compliant wireless device via said web server, Internet
and WAP gateway wherein said display definition includes instructions for generating
WML code to display static text and WAP bitmap images on said WAP compliant
wireless device; said menu selection definition includes instructions for generating
WML code to display a list of menu items as hyperlinks on said WAP compliant
15 wireless device; said option selection definition includes instructions for generating
WML code to display option items on said WAP compliant wireless device; and said
user input definition includes instructions for generating WML code to obtain user
input from said WAP compliant wireless device and wherein said retrieve data
definition includes instructions for generating WML code to retrieve data from a data
20 source; said insert data definition includes instructions for generating WML code to
insert data to a data source; said update data definition includes instructions for
generating WML code to update data in a data source; and said delete data definition
includes instructions for generating WML code to delete data from a data source.

According to still yet another aspect of the present invention there is
25 provided a system for creating and conveying a wireless application to a wireless
device comprising:

a wireless device generating a wireless application request;
a gateway generating a web request in response to the wireless
application request and conveying the web request to a web server over the Internet;
30 a server reading and executing application definitions from a database
that are appropriate for generating executable code forming a wireless application

-4c-

response in response to said wireless application request and dynamically generating the executable code forming said wireless application response using the application definitions read from said database, said server transmitting said wireless application response to said wireless device via said web server, Internet and gateway; and

5 a wireless application builder creating wireless applications in response to user input and storing the application definitions making up each built wireless application in said database, during building of a wireless application, said wireless application builder providing a visual wireless application creation platform thereby to enable said user to define and view the application definitions making up the wireless

10 application being built, in graphical logic flow form thereby to shield said user from protocol detail, wherein said application definitions define action steps in the wireless application being built and comprise user interface, data and logical flow control definitions.

According to still yet another aspect of the present invention there is

15 provided a method for building a wireless application and conveying the built wireless application to a wireless device comprising:

at design time:

presenting a graphical user interface defining a visual wireless application creation platform;

20 creating wireless application definitions corresponding to a graphical logic flow diagram modeling said wireless application being built and displayed by said graphical user interface in windows in response to user input thereby to shield said user from protocol detail, wherein said wireless application definitions define action steps in the wireless application being built and comprise

25 user interface, data and logical flow control definitions; and

storing the created application definitions in a database; and

at run time:

executing the wireless application definitions making up the built wireless application in response to a wireless device request thereby to generate

30 dynamically executable code forming said wireless application; and

transmitting the executable code to said wireless device for execution.

The present invention provides advantages in that WAP applications can be developed quickly, easily and cost-effectively. This is achieved by providing a graphical user interface that allows WAP application developers to define WAP applications easily using intuitive definitions thereby shielding WAP application developers from the technical complexities of WAP protocols and standards. In this manner, WAP application developers are able to focus on the development of WAP applications.

Brief Description of the Drawings

10 An embodiment of the present invention will now be described more fully with reference to the accompanying drawings in which:

Figure 1 is a diagrammatic overview of the conventional manner by which a WAP compliant wireless device accesses a WAP application over the Internet;

15 Figure 2 is a diagrammatic overview of the manner by which a WAP compliant wireless device accesses a WAP application over the Internet in accordance with the present invention;

Figure 3 is a flow diagram showing an exemplary WAP application including a plurality of application definitions;

20 Figure 4 is a schematic block diagram of a WAP builder used to construct application definitions for WAP applications in accordance with the present invention; and

Figures 5a to 5000 are screen shots showing the steps performed during creation of application definitions for a stock quote WAP application.

25

Detailed Description of the Preferred Embodiment

The present invention relates generally to a system and method for generating and deploying WAP applications specified by user intuitive application definitions through a graphical user interface. The system includes two primary components, namely a WAP application builder and a WAP server. The WAP application builder allows a user to define application definitions for a WAP application and to view the WAP application flow in graphical form. The application

definitions, which define action steps in the WAP application, are stored in a database. When the database of application definitions is deployed, the WAP server uses the application definitions to create WML code necessary to generate WAP responses in response to WAP requests received from a WAP compliant wireless device. The WML code is generated dynamically and is delivered to the WAP compliant wireless device over the Internet via conventional web servers. Specifics of the present invention will now be described with reference to the Figures.

For ease of understanding, the conventional manner by which a WAP compliant wireless device accesses a WAP application over the Internet will firstly be described with reference to Figure 1. As can be seen, when a WAP compliant wireless device 102 generates a WAP request for an Internet accessible application, the WAP request is forwarded to a WAP gateway 104. The WAP gateway 104 provides an interface between the WAP compliant wireless device 102 and the Internet 106 and supports the overhead necessary for the WAP compliant wireless device to access the Internet 106. The WAP gateway 104 in turn forwards the WAP request, in HTTP format, to a web server 108 hosting the Internet accessible application 110 of interest via the Internet 106. If the Internet accessible application 110 is WAP compliant (i.e. it has WAP request responses pre-coded in WML), the web server 108 returns a WAP response comprised of a deck of WML cards, to the WAP gateway 104 via the Internet 106. The WAP gateway 104 in turn passes the WAP response to the WAP compliant wireless device 102. The browser running on the WAP compliant wireless device 102 executes the WML cards of the WAP response thereby to display the Internet accessible application on its display screen.

If the Internet accessible application 110 is not WAP compliant, the web server 108 returns HTML web pages to the WAP gateway 104 via the Internet 106 in response to the WAP request. The WAP gateway 104, to the extent that it is able, translates the HTML web pages into WML cards before transmitting the WML cards to the WAP compliant wireless device 102. Unfortunately, this process is typically unworkable due to the limitations associated with WAP compliant wireless devices described previously.

Figure 2 illustrates the manner by which a WAP compliant wireless device accesses a WAP compliant Internet accessible application over the Internet in

accordance with the present invention. Similar to the conventional method, when the WAP compliant device 202 generates a WAP request for an Internet accessible WAP application, the WAP request is forwarded to a WAP gateway 204. The WAP gateway 204 in turn forwards a web request to the appropriate web server 208 via the Internet 206. Web server 208 in turn passes a WAP request to a WAP server 212. WAP server 212, in response to the WAP request, reads and interprets application definitions that are specified in a logical order from a database 214 that are appropriate for a WAP response.

The WAP request delivered to the WAP server 212 by the web server 208 is in the form of a Uniform Resource Locator (URL) that includes additional parameters used to pass information to the WAP server 212. This additional information includes, for example, application specific data such as user input data and directions with respect to the next application definition to be processed. The information may further include capability particulars of the WAP compliant wireless device 202 so that the WAP server 212 may generate a WAP response that takes full advantage of the capabilities of the WAP compliant wireless device 202.

The application definitions in the database 214 are created and stored at design time, and are interpreted by the WAP server 212 at run time to dynamically generate WML code in response to the WAP request as will be described. Each application definition in the database 214 defines an action step in the WAP application.

The application definitions are grouped into a plurality of categories. At least one of the application definitions is designated as an anchor and is executed first whenever a WAP request for the WAP application is received. In the present embodiment, the application definitions are grouped into one of three different categories, namely a user interface definitions category, a data definitions category and a flow control definitions category. Each application definition category includes different types of application definitions, with each application definition being designed for a specific purpose.

Table 1 below sets out the application definitions in the user interface definitions category and provides a description of the application definitions in the category and lists the parameters associated with the application definitions. As can

be seen, this category includes a display definition, a menu selection definition, an option selection definition and a user input definition. The display definition is designed to display static text or WAP bitmap images on a WAP compliant wireless device. The menu selection definition is designed to display a list of menu items in the form of hyperlinks on a WAP compliant wireless device. The option selection definition is designed to display a list of option items on a WAP compliant wireless device. The user input definition is designed to obtain user input received from a WAP compliant wireless device.

10

Table 1: User Interface Definitions Category

Type	Description	Parameters
Display	Display static text or WAP bitmap image on WAP compliant wireless device. Text may be displayed in different formats: different font sizes, underlined, bolded and italicized.	Static text with formatting information (such as bold, underline, italic, etc); WAP Bitmap image file; and Next application definition.
Menu Selection	Display a list of selections for a user to choose from. The selections are implemented as a series of Hyper Links.	Text of the list of menu items; and Action (i.e., next application definition) for each menu item.
Option Selection	Display a list of selections for a user to choose from. Multiple selections are allowed.	Text of the list of option items Variable name for storing selected options; and Next application definition
User Input	Obtain user input from WAP compliant wireless device.	Variable name for storing input data; Default value; Input type (text or password); Optional input only; Maximum input length; and Next application definition

Table 2 below sets out the application definitions in the data definitions category. Similarly, a description of the application definitions in the category and the parameters associated with the application definitions are included. As can be seen, this category includes a retrieve data definition, an insert data

15

definition, an update data definition and a delete data definition. The retrieve data definition is designed to retrieve data from a data source. The insert data definition is designed to insert data into a data source. The update data definition is designed to update data in a data source. The delete data definition is designed to delete data from a data source.

Table 2: Data Definitions Category

Type	Description	Parameters
Retrieve Data	Retrieve data from data source	Data source specification; Login name and password of the data source; Maximum number of records to be returned; Sorting order of records; Language encoding format of the data; Retrieval criteria; Record fields to be retrieved and displayed; and Action (i.e., next application definition) for success versus failure.
Insert Data	Insert data to data source	Data source specification; Login name and password of the data source; Language encoding format of the data; Data fields information; and Action (i.e., next application definition) for success versus failure.
Update Data	Update data to data source	Data source specification; Login name and password of the data source; Language encoding format of the data; Criteria of records to be updated; Data fields to be updated; and Action (i.e., next application definition) for success versus failure.
Delete Data	Delete data from data source	Data source specification; Login name and password of the data source; Language encoding format of the data; Deletion criteria; and Action (i.e., next application definition) for success versus failure.

Table 3 below sets out the application definitions in the flow control definitions category. Again a description of the application definitions in the category and the parameters associated with the application definitions are included. As can
 5 been seen, this category includes an If-Then-Else definition and a function call definition. The If-Then-Else definition is designed to define an "If-Then-Else" logic construct. The function call definition is designed to execute a user defined function.

Table 3: Flow Control Definitions Category

10

Type	Description	Parameters
IF-THEN-ELSE	Define an "if-then-else" logic construct in the form of: IF <i>variable operator value</i> THEN JUMP TO <i>xx</i> ELSE JUMP TO <i>yy</i>	Condition variable name; Condition operator; Condition value; Condition value type; Case sensitivity comparison; and Next application definition.
Function Call	Execute a user defined function	Function call Name; and Parameter list.

The WAP server 212 interprets the application definitions read from the database 214, interacts with Internet websites 216 as required for appropriate data and dynamically creates the WML code for a WAP response. Once created, the WAP
 15 response is transmitted to the WAP compliant wireless device 202 via web server 208, Internet 206 and WAP gateway 204. The browser running on the WAP compliant wireless device 202 executes the WAP response thereby to display the WAP application on its display screen.

During the reading and interpreting of application definitions from the
 20 database 214, the WAP server 212 enhances the transfer of WML code from the WAP server 212 to the web server by examining the application definitions of the WAP application ahead of time. In this look-ahead process, one or more application definitions are combined to form a deck of WML cards. During the look-ahead process, the WAP server 212 combines as many consecutive application definitions in
 25 the user interface definitions category as possible to form a deck of WML cards, but

stops the look-ahead process when an application definition in the data definitions category or the flow control definitions category is encountered.

Figure 3 shows a flow diagram of a WAP application designed to retrieve a price and quantity record as well as a customer record. As should be appreciated, this example is provided for the purpose of illustration only and is not intended to limit the scope of the present invention. The WAP application is constructed using a series of application definitions specified in a logical manner. In this particular example, the application definitions include display definitions, user input definitions, retrieve data definitions and menu selection definitions.

As mentioned previously, the application definitions that are used by the WAP server to generate WAP responses are created at design. In order to create the application definitions, a WAP application builder 400 executed by a computer is provided. Figure 4 is a schematic block diagram of the WAP application builder and as can be seen, WAP builder 400 includes a graphical user interface (GUI) 402, a WAP generator 404, and a library database 406. During WAP application development, a WAP application developer defines the application definitions through GUI 402. The GUI 402 displays the application definitions in graphical form for ease of use in defining the flow of steps. The library database 406 contains a library of all of the different application definitions in each of the application definition categories.

When a WAP application is to be deployed, the appropriate application definitions making up the WAP application are stored in the database 214 that is accessed by the WAP server 212 in response to WAP requests.

The graphical user interface 402 presents different windows to a WAP application developer that allows the WAP application developer to generate different application definitions by completing the fields of these windows. Once completed, an application definition is created that can be linked to other application definition through a graphical window that presents the logical flow of the WAP application. Upon completion of the graphically defined application definitions, the WAP generator creates WAP application definition data (instructions) for creating WML code to carry out the defined application definitions.

An example of the manner by which a WAP application is created using the WAP application builder will now be described with particular reference to Figures 5a to 5ooo. In this particular example, a WAP application for stock quotes is described; however those of skill in the art will appreciate that this description is for the purpose of illustration only and is not intended to limit the scope of the present invention.

Turning now to Figure 5a, an icon representing the WAP application builder 400 is shown as displayed on the monitor of a computer (not shown). Selection of the icon 500 involves the WAP application builder opens a window that provides the user with a number of options including an "Add new WAP application" option as shown in Figure 5b. Selecting the "Add new WAP application" option and clicking on the Next button opens a main window as shown in Figure 5c. The main window includes a name field and a description field to allow the user to name and describe the new WAP application being created.

Following entry of a name and a short description for the WAP application being created into the appropriate fields, the user must click on the display icon of the main window toolbar, click inside the main window, drag the cursor outward and then release the mouse button. When this is done, a display window opens as shown in Figure 5d. As can be seen, the display window includes a task name field, a task title field and a task pop-up field.

Following entry of the appropriate information into the fields of the display window as shown in Figure 5e and upon clicking of the "OK" button, a WELCOME task box is created and displayed in the main window as shown in Figure 5f. Clicking the input icon on the main window toolbar, clicking inside the main window, dragging the cursor outward and then releasing the mouse button causes an input window to open as shown in Figure 5g. As can be seen, the input window includes a prompt field, a task pop-up tip field, a default value field, an input format field and a maximum length field.

Following entry of the appropriate information into the fields of the input window as shown in Figure 5h and upon clicking of the "OK" button, a SYMBOL task box is created and displayed in the main window below the WELCOME task box as shown in Figure 5i. Clicking on the line icon on the main

window toolbar, clicking inside the WELCOME task box and then extending the line into the SYMBOL task box joins the WELCOME task box to the SYMBOL task box as shown in Figure 5j.

Following this, a display menu window is opened by clicking on the menu items icon on the main window toolbar, clicking inside the main window, 5 dragging the cursor outward and then releasing the mouse button as shown in Figure 5k. As can be seen the display menu includes a task name field, a task title field and a task pop-up tip field as well as menu add, edit and remove buttons. After entry of the appropriate information into the fields of the display menu window, an add menu 10 window is opened by clicking on the add button as shown in Figure 5l. The add menu window includes a hint field, a display text field, an image field and an image alt text field. Figure 5l shows the word "Hint" entered in the hint field and the word "Quote" entered in the display text field. Figure 5m shows another add menu window with the word "Hint" entered in the hint field and the word "News" entered in the display text 15 field.

Once the add menu windows of Figures 5l and 5m have been closed by clicking on the OK buttons, the display menu window is updated to show the added menus as shown in Figure 5n. The task name field and the task pop-up tip fields are shown with entries. Clicking on the OK button of the display menu window, creates a 20 MAINMENU task box that is displayed in the main window below the SYMBOL task box as shown in Figure 5o. Clicking on the line icon on the main window toolbar, clicking inside the SYMBOL task box and then extending the line into the MAINMENU task box joins the SYMBOL task box to the MAINMENU task box as shown in Figure 5p.

25 A data retrieval-display data window is then opened by clicking on a display data icon on the main window toolbar, clicking inside the main window, dragging the cursor outward and then releasing the mouse button as shown in Figure 5q. The appropriate information is then entered into the fields of the data retrieval-display data window as shown in Figure 5r.

30 Selecting the fields tab within the data retrieval-display data window exposes an add button, an edit button and a remove button as shown in Figure 5s. These buttons permit fields to be modified. Clicking on the add button opens an add

-14-

field information window as shown in Figure 5t. A Name field is then added by completing the fields of the field information window as shown in Figure 5u. Once the fields are completed and the OK button is selected, an appropriate field entry is created and appears under the field tab of the display data window as shown in Figure 5v. In this particular example, Last, Change, Bid, and Ask field entries also appear under the field tab in Figure 5v. These field entries are created in the same manner as the Name field entry. The information entered into the fields of the add field information window to create the Last, Change, Bid and Ask field entries is shown in Table 4 below.

10

Table 4

Field Name	Header	Type	Visible
Last	Lst: 	Numeric	Yes
Change	Chg: 	Numeric	Yes
Bid	Bid: 	Numeric	Yes
Ask	Ask: 	Numeric	Yes

Criteria is then entered into the criteria field of the data retrieval-display data window. The advance tab is then selected in the data retrieval-display data window so that the ADO connection string can be completed as shown in Figure 5w. The messages tab is then selected to expose the messages fields and the appropriate information is entered into these fields as shown in Figure 5x. With the above steps performed, the data retrieval-display data window is closed by clicking on the OK button. In response, a QUOTE task box is displayed in the main window below the MAINMENU task box as shown in Figure 5y. Clicking on the line icon on the main window toolbar, clicking inside the MAINMENU task box and then extending the line into the QUOTE task box joins the MAINMENU task box to the QUOTE task box as shown in Figure 5z. To link the QUOTE task box back to the MAINMENU task box, a link task box icon is selected, a click is made inside the QUOTE task box, and a line is extended into the MAINMENU task box as shown in Figure 5aa.

Following this, the display menu window is opened again by clicking on the menu items icon on the main window toolbar, clicking inside the main

window, dragging the cursor outward and then releasing the mouse button as shown in Figure 5bb.

Add menu windows are then opened by clicking on the add button as shown in Figure 5cc. A "Detailed Quote" menu and a "Next Symbol" menu are
5 created by completing the add menu windows as shown in Figures 5cc and 5dd. Once completed, the Detailed Quote menu and Next Symbol menu appear in the display menu window as shown in Figure 5ee.

Clicking on the OK button of the display menu window creates a MENU2 task box that is displayed in the main window as shown in Figure 5ff.
10 Clicking on the line icon on the main window toolbar, clicking inside the QUOTE task box and then extending the line into the MENU2 task box joins the QUOTE task box to the MENU2 task box as shown in Figure 5gg. To link the MENU2 task box back to the SYMBOL task box, the link task box icon is selected, a click is made inside the MENU2 task box, and a line is extended into the SYMBOL task box as
15 shown in Figure 5hh.

The data retrieval-display data window is then opened again by clicking on the display data icon on the main window toolbar, clicking inside the main window, dragging the cursor outward and then releasing the mouse button as shown in Figure 5ii. The appropriate information is then entered into the fields of the data
20 retrieval-display data window as shown in Figure 5jj.

Selecting the fields tab within the data retrieval-display data window exposes the add button, the edit button and the remove button as shown in Figure 5kk. Clicking on the add button opens the add field information window as shown in Figure 5ll. A Name field is then added by completing the fields of the field
25 information window as shown in Figure 5mm. Once the fields are completed and the OK button is selected, an appropriate field entry is created and appears under the field tab of the data retrieval-display data window as shown in Figure 5nn. In this particular example, Volume, DayHi, DayLo, YearHi, and YearLo field entries also appear under the field tab in Figure 5nn. These field entries are created in the same
30 manner as the Name field entry. The information entered into the fields of the add field information window to create the Volume, DayHi, DayLo, YearHi, and YearLo field entries is shown in Table 5 below.

Table 5

Field Name	Header	Type	Visible
Volume	Vol: 	Numeric	Yes
DayHi	Dhi: 	Numeric	Yes
DayLo	Dlo: 	Numeric	Yes
YearHi	Yhi: 	Numeric	Yes
YearLo	Ylo: 	Numeric	Yes

Criteria is then entered into the criteria field of the data retrieval-
5 display data window. The advance tab is then selected in the data retrieval-display
data window so that the ADO connection string can be completed as shown in Figure
500. The messages tab is then selected to expose the messages fields and the
appropriate information is entered into these fields as shown in Figure 5pp. With the
above steps performed, the data retrieval-display data window is closed by clicking on
10 the OK button. In response, a DETAILQUOTE task box is displayed in the main
window below the MENU2 task box as shown in Figure 5qq. Clicking on the line
icon on the main window toolbar, clicking inside the MENU2 task box and then
extending the line into the DETAILQUOTE task box joins the MENU2 task box to
the DETAILQUOTE task box as shown in Figure 5rr. To link the DETAILQUOTE
15 task box back to the SYMBOL task box, the link task box icon is selected, a click is
made inside the DETAILQUOTE task box, and a line is extended into the SYMBOL
task box as shown in Figure 5ss.

The data retrieval-display data window is then opened by clicking on
the display data icon on the main window toolbar, clicking inside the main window,
20 dragging the cursor outward and then releasing the mouse button as shown in Figure
5tt. The appropriate information is then entered into the fields of the data retrieval-
display data window as shown in Figure 5uu.

Selecting the fields tab within the data retrieval-display data window
exposes the add button, the edit button and the remove button as shown in Figure 5vv.
25 Clicking on the add button opens the add field information window as shown in
Figure 5ww. A RecordID field is then added by completing the fields of the field
information window as shown in Figure 5xx. Once the fields are completed and the

OK button is selected, an appropriate field entry is created and appears under the field tab of the data retrieval-display data window as shown in Figure 5yy. In this particular example, a Headline field entry also appears under the field tab in Figure 5yy. This field entry is created in the same manner as the Headline field entry.

5 Criteria is then entered into the criteria field of the data retrieval-display data window as shown in Figure 5zz. The advance tab is then selected in the data retrieval-display data window so that the ADO connection string can be completed as shown in Figure 5aaa. The messages tab is then selected to expose the messages fields and the appropriate information is entered into these fields as shown
10 in Figure 5bbb. With the above steps performed, the data retrieval-display data window is closed by clicking on the OK button. In response, a HEADLINE task box is displayed in the main window as shown in Figure 5ccc. Clicking on the line icon on the main window toolbar, clicking inside the MAINMENU task box and then extending the line into the HEADLINE task box joins the MAINMENU task box to
15 the HEADLINE task box as shown in Figure 5ddd.

The data retrieval-display data window is then opened yet again by clicking on the display data icon on the main window toolbar, clicking inside the main window, dragging the cursor outward and then releasing the mouse button as shown in Figure 5eee. The appropriate information is then entered into the fields of the
20 display data window as shown in Figure 5fff.

Selecting the fields tab within the data retrieval-display data window exposes the add button, the edit button and the remove button as shown in Figure ggg. Clicking on the add button opens the add field information window as shown in Figure 5hhh. A Detail field is then added by completing the fields of the field
25 information window as shown in Figure 5iii. Once the fields are completed and the OK button is selected, an appropriate field entry is created and appears under the field tab of the data retrieval-display data window as shown in Figure 5jjj.

Criteria is then entered into the criteria field of the data retrieval-display data window as shown in Figure 5jjj. The advance tab is then selected in the
30 data retrieval-display data window so that the ADO connection string can be completed as shown in Figure 5kkk. The messages tab is then selected to expose the messages fields and the appropriate information is entered into these fields as shown

-18-

in Figure 5lll. With the above steps performed, the data retrieval-display data window is closed by clicking on the OK button. In response, a DETAILNEWS task box is displayed in the main window as shown in Figure 5mmm. Clicking on the line icon on the main window toolbar, clicking inside the HEADLINE task box and then
5 extending the line into the DETAILNEWS task box joins the HEADLINE task box to the DETAILNEWS task box as shown in Figure 5nnn. To link the DETAILNEWS task box back to the SYMBOL task box, the link task box icon is selected, a click is made inside the DETAILNEWS task box, and a line is extended into the SYMBOL task box as shown in Figure 5ooo.

10 By completing the above steps, a stock quote WAP application is generated based on a number of application definitions created through the graphical user interface 402. As will be appreciated, the graphical user interface provides a commercial facility for WAP application developers to create the application definitions while avoiding technical details associated with WAP specifications.

15 Although a preferred embodiment of the present invention has been described, it will be understood by those skilled in the art that variations and modifications may be made thereto without departing from the spirit of the invention or the scope of the appended claims.

-19-

What is claimed is:

1. A system for creating and conveying a wireless application to a wireless device comprising:
 - 5 a wireless device generating a wireless application request;
 - a gateway generating a web request in response to the wireless application request and conveying the web request to a web server over the Internet;
 - a server reading application definitions from a database that are appropriate for generating a wireless application response, in response to said wireless application request and dynamically generating code forming said wireless application response using the application definitions read from said database, said server transmitting said wireless application response to said wireless device via said web server, Internet and gateway; and
 - 10 a wireless application builder creating wireless applications in response to user input and storing the application definitions making up the logic and interfaces of each built wireless application in said database, during building of a wireless application, said wireless application builder providing a visual wireless application creation platform thereby to enable said user to define and view the application definitions making up the wireless application being built, in graphical logic flow form.
 - 15
 - 20
2. A system according to claim 1 wherein said server further accesses Internet sites for data during generation of said code.
- 25 3. A system according to claim 2 wherein each of said application definitions includes a description and parameters, said parameters defining information associated with said application definitions.
- 30 4. A system according to claim 3 wherein said application definitions are grouped into categories.

-20-

5. A system according to claim 4 wherein said categories include a user interface definitions category, a data definitions category and a flow control definitions category.

5 6. A system according to claim 5 wherein said user interface definitions category includes a display definition, a menu selection definition, an option selection definition and a user input definition.

7. A system according to claim 5 or 6 wherein said data definitions category includes a retrieve data definition, an insert data definition, an update definition and a delete data definition.

8. A system according to any one of claims 5 to 7 wherein said flow control definitions include an If-Then-Else definition and a function call definition.

15

9. A system according to claim 6 wherein said display definition includes instructions for generating code to display static text and bitmap images on said wireless device; said menu selection definition includes instructions for generating code to display a list of menu items as hyperlinks on said wireless device; said option selection definition includes instructions for generating code to display option items on said wireless device; and said user input definition includes instructions for generating code to obtain user input from said wireless device.

20

10. A system according to claim 7 wherein said retrieve data definition includes instructions for generating code to retrieve data from a data source; said insert data definition includes instructions for generating code to insert data to a data source; said update data definition includes instructions for generating code to update data in a data source; and said delete data definition includes instructions for generating code to delete data from a data source.

30

11. A system according to claim 8 wherein said If-Then-Else definition includes instructions for generating code to create an if-then-else logic construct and

-21-

said function call definition includes instructions for generating code to call a subroutine.

12. A wireless application builder for creating a wireless application
5 comprising:

a graphical user interface providing a visual wireless application platform enabling a user to define and view application definitions making up a wireless application being built in graphical logic flow form; and

10 a wireless application generator creating application definition data defining the logic and interfaces of the wireless application being built in response to defined application definitions, wherein said graphical user interface presents windows for said application definitions.

13. A wireless application builder according to claim 12 further
15 comprising a database storing said application definitions, said application definitions being grouped into categories including a user interface definitions category, a data definitions category and a flow control definitions category.

14. A wireless application builder according to claim 13 wherein said user
20 interface definitions category includes a display definition, a menu selection definition, an option selection definition and a user input definition.

15. A wireless application builder according to claim 14 wherein said data
25 definitions category includes a retrieve data definition, an insert data definition, an update definition and a delete data definition.

16. A wireless application builder according to claim 15 wherein said flow control definitions include an If-Then-Else definition and a function call definition.

30 17. A wireless application builder according to any one of claims 14 to 16 wherein said display definition includes instructions for generating code to display static text and bitmap images on said wireless device; said menu selection definition includes instructions for generating code to display a list of menu items as hyperlinks

-22-

on said wireless device; said option selection definition includes instructions for generating code to display option items on said compliant wireless device; and said user input definition includes instructions for generating code to obtain user input from said wireless device.

5

18. A wireless application builder according to claim 15 wherein said retrieve data definition includes instructions for generating code to retrieve data from a data source; said insert data definition includes instructions for generating code to insert data to a data source; said update data definition includes instructions for generating code to update data in a data source; and said delete data definition includes instructions for generating code to delete data from a data source.

10

19. A wireless application builder according to claim 16 wherein said If-Then-Else definition includes instructions for generating code to create an if-then-else logic construct and said function call definition includes instructions for generating code to call a subroutine.

15

20. A method for building a wireless application comprising the steps of:
presenting a graphical user interface defining a visual wireless application creation platform;
defining wireless application definitions corresponding to a graphical logic flow diagram modeling said wireless application in response to user input;
presenting said application definitions in windows; and
creating application definition data in response to defined application definitions, said application definition data defining the logic and interfaces of the wireless application being built.

20

25

21. A method of conveying a wireless application protocol (WAP) application to a WAP compliant wireless device comprising the steps of:
generating a WAP request using a WAP compliant wireless device;
conveying the WAP request to a WAP gateway;
at the WAP gateway generating a web request in response to the WAP request and conveying the web request to a web server over the Internet;

30

-23-

at the web server generating a WAP request in response to the web request and conveying the WAP request to a WAP server;

at said WAP server reading application definitions from a database that are appropriate for generating a WAP response in response to said WAP request and
5 dynamically generating wireless markup language (WML) code forming said WAP response using the application definitions read from said database, said application definitions comprising display, menu selection, option selection, user input, retrieve data, insert data, update data and delete definitions; and

transmitting said WAP response to said WAP compliant wireless
10 device via said web server, Internet and WAP gateway wherein said display definition includes instructions for generating WML code to display static text and WAP bitmap images on said WAP compliant wireless device; said menus selection definition includes instructions for generating WML code to display a list of menu items as
15 hyperlinks on said WAP compliant wireless device; said option selection definition includes instructions for generating WML code to display option items on said WAP compliant wireless device; and said user input definition includes instructions for generating WML code to obtain user input from said WAP compliant wireless device and wherein said retrieve data definition includes instructions for generating WML
20 code to retrieve data from a data source; said insert data definition includes instructions for generating WML code to insert data to a data source; said update data definition includes instructions for generating WML code to update data in a data source; and said delete data definition includes instructions for generating WML code to delete data from a data source.

25 22. The method of claim 21 further comprising the step of accessing data from Internet sites during said WML code generating.

23. The method of claim 22 wherein said WAP request includes information concerning the capabilities of said WAP compliant wireless device.

30

24. A system for conveying a wireless application protocol (WAP) application to a WAP compliant wireless device comprising:

a WAP compliant wireless device generating a WAP request;

-24-

a WAP gateway generating a web request in response to the WAP request and conveying the web request to a web server over the Internet; and

a WAP server reading application definitions from a database that are appropriate for generating a WAP response in response to said WAP request and
5 dynamically generating wireless markup language (WML) code forming said WAP response using the application definitions read from said database, said application definitions comprising display, menu selection, option selection, user input, retrieve data, insert data, update data and delete definitions, said WAP server transmitting said WAP response to said WAP compliant wireless device via said web server, Internet
10 and WAP gateway wherein said display definition includes instructions for generating WML code to display static text and WAP bitmap images on said WAP compliant wireless device; said menu selection definition includes instructions for generating WML code to display a list of menu items as hyperlinks on said WAP compliant wireless device; said option selection definition includes instructions for generating
15 WML code to display option items on said WAP compliant wireless device; and said user input definition includes instructions for generating WML code to obtain user input from said WAP compliant wireless device and wherein said retrieve data definition includes instructions for generating WML code to retrieve data from a data source; said insert data definition includes instructions for generating WML code to
20 insert data to a data source; said update data definition includes instructions for generating WML code to update data in a data source; and said delete data definition includes instructions for generating WML code to delete data from a data source.

25. A system according to claim 24 wherein said WAP server further
25 accesses Internet sites for data during generation of said WML code.

26. A system according to claim 25 wherein each of said application definitions includes a description and parameters, said parameters defining information associated with said application definitions.
30

27. A system according to claim 26 wherein said application definitions are grouped into categories.

-25-

28. A system for creating and conveying a wireless application to a wireless device comprising:
- a wireless device generating a wireless application request;
 - a gateway generating a web request in response to the wireless application request and conveying the web request to a web server over the Internet;
 - a server reading and executing application definitions from a database that are appropriate for generating executable code forming a wireless application response in response to said wireless application request and dynamically generating the executable code forming said wireless application response using the application definitions read from said database, said server transmitting said wireless application response to said wireless device via said web server, Internet and gateway; and
 - a wireless application builder creating wireless applications in response to user input and storing the application definitions making up each built wireless application in said database, during building of a wireless application, said wireless application builder providing a visual wireless application creation platform thereby to enable said user to define and view the application definitions making up the wireless application being built, in graphical logic flow form thereby to shield said user from protocol detail, wherein said application definitions define action steps in the wireless application being built and comprise user interface, data and logical flow control definitions.
29. A system according to claim 28 wherein said server further accesses Internet sites for data during generation of said code.
30. A system according to claim 29 wherein each of said application definitions includes a description and parameters, said parameters defining information associated with said application definitions.
31. A system according to claim 30 wherein said application definitions are grouped into categories.

-26-

32. A system according to claim 31 wherein said categories include a user interface definitions category, a data definitions category and a flow control definitions category.

5 33. A system according to claim 32 wherein user interface definitions in said user interface definitions category includes a display definition, a menu selection definition, an option selection definition and a user input definition.

10 34. A system according to claim 32 or 33 wherein data definitions in said data definitions category includes a retrieve data definition, an insert data definition, an update definition and a delete data definition.

15 35. A system according to any one of claims 32 to 34 wherein flow control definitions in said flow control definitions categories include an If-Then-Else definition and a function call definition.

20 36. A system according to claim 33 wherein said display definition includes instructions for generating code to display static text and bitmap images on said wireless device; said menus selection definition includes instructions for generating code to display a list of menu items as hyperlinks on said wireless device; said option selection definition includes instructions for generating code to display option items on said wireless device; and said user input definition includes instructions for generating code to obtain user input from said wireless device.

25 37. A system according to claim 34 wherein said retrieve data definition includes instructions for generating code to retrieve data from a data source; said insert data definition includes instructions for generating code to insert data to a data source; said update data definition includes instructions for generating code to update data in a data source; and said delete data definition includes instructions for
30 generating code to delete data from a data source.

38. A system according to claim 35 wherein said If-Then-Else definition includes instructions for generating code to create an if-then-else logic construct and

-27-

said function call definition includes instructions for generating code to call a subroutine.

39. A system according to any one of claims 28 to 38 wherein said server
5 combines consecutive application definitions of a selected type and transmits the executable code associated with the combined application definitions to said wireless device to facilitate transmission of said executable code.

40. A system according to claim 39 wherein said selected application
10 definition type comprises user interface definitions.

41. A system according to any one of claims 28 to 40 wherein one of said
application definitions is designated as an anchor, said anchor application definition
being executed by said server first.

15

42. A method for building a wireless application and conveying the built
wireless application to a wireless device comprising:

at design time:

20 presenting a graphical user interface defining a visual wireless application creation platform;

creating wireless application definitions corresponding to a graphical logic flow diagram modeling said wireless application being built and displayed by said graphical user interface in windows in response to user input thereby to shield said user from protocol detail, wherein said wireless application
25 definitions define action steps in the wireless application being built and comprise user interface, data and logical flow control definitions; and

storing the created application definitions in a database; and

at run time:

30 executing the wireless application definitions making up the built wireless application in response to a wireless device request thereby to generate dynamically executable code forming said wireless application; and

transmitting the executable code to said wireless device for execution.

-28-

43. The method of claim 42 wherein one of said application definitions is designated as an anchor, during said executing, said anchor application definition being executed first.

5

44. The method of claim 42 or 43 wherein prior to transmitting, consecutive application definitions of a selected type are combined, the executable code associated with the combined application definitions being transmitted to said wireless device.

10

45. The method of claim 44 wherein said selected application definition type comprises user interface definitions.

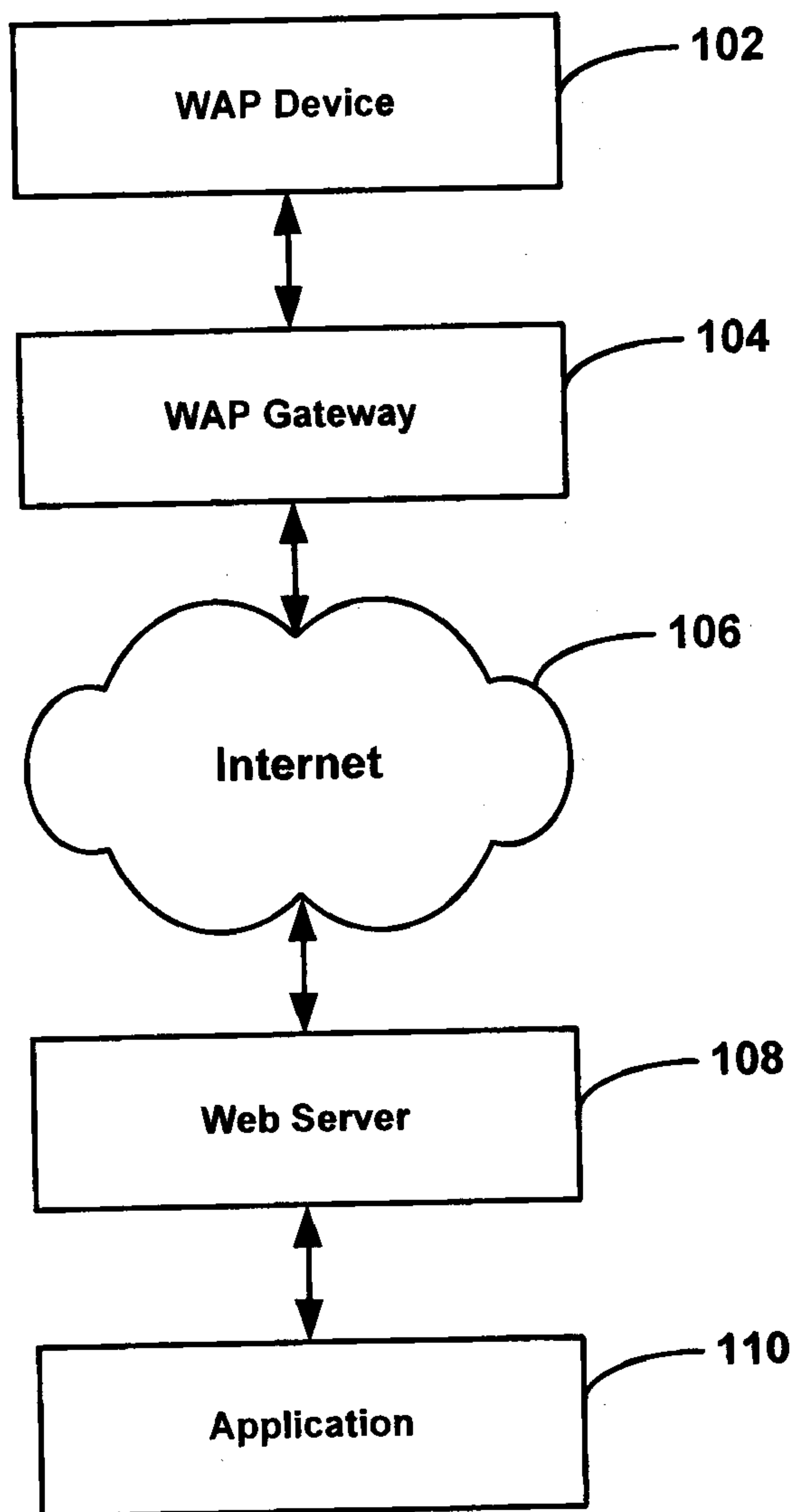


Fig. 1
(Prior Art)

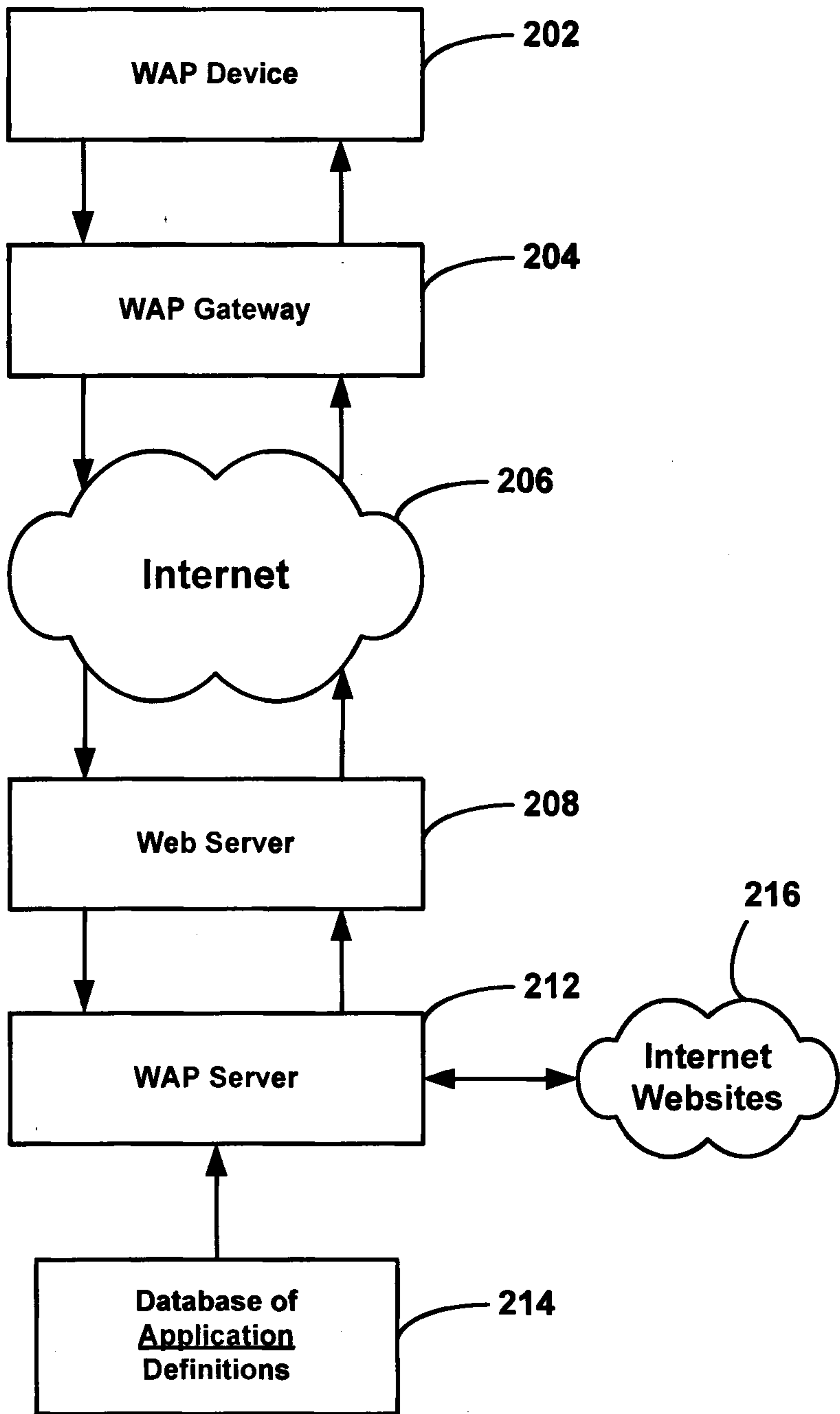


Fig. 2

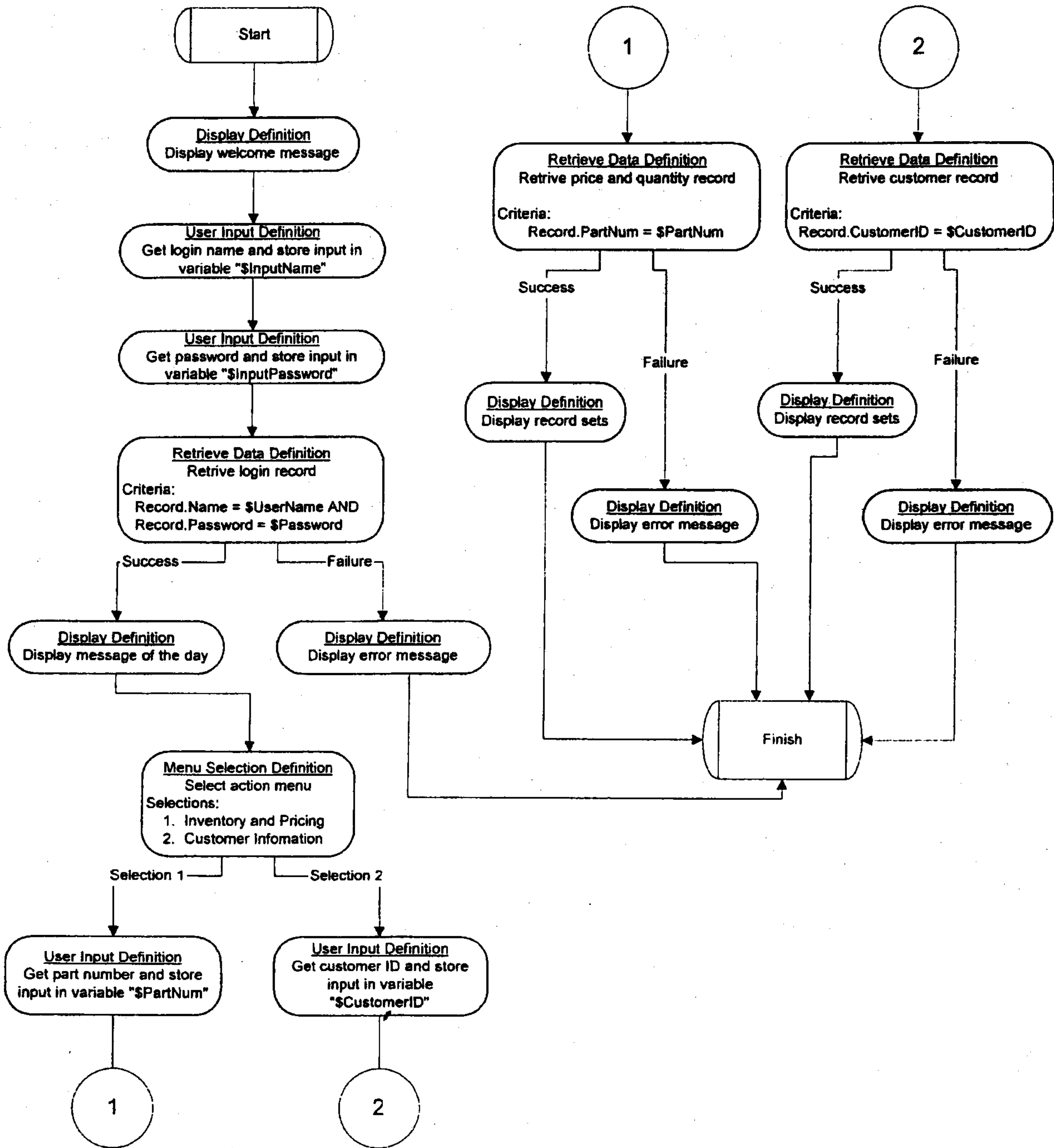


Figure 3

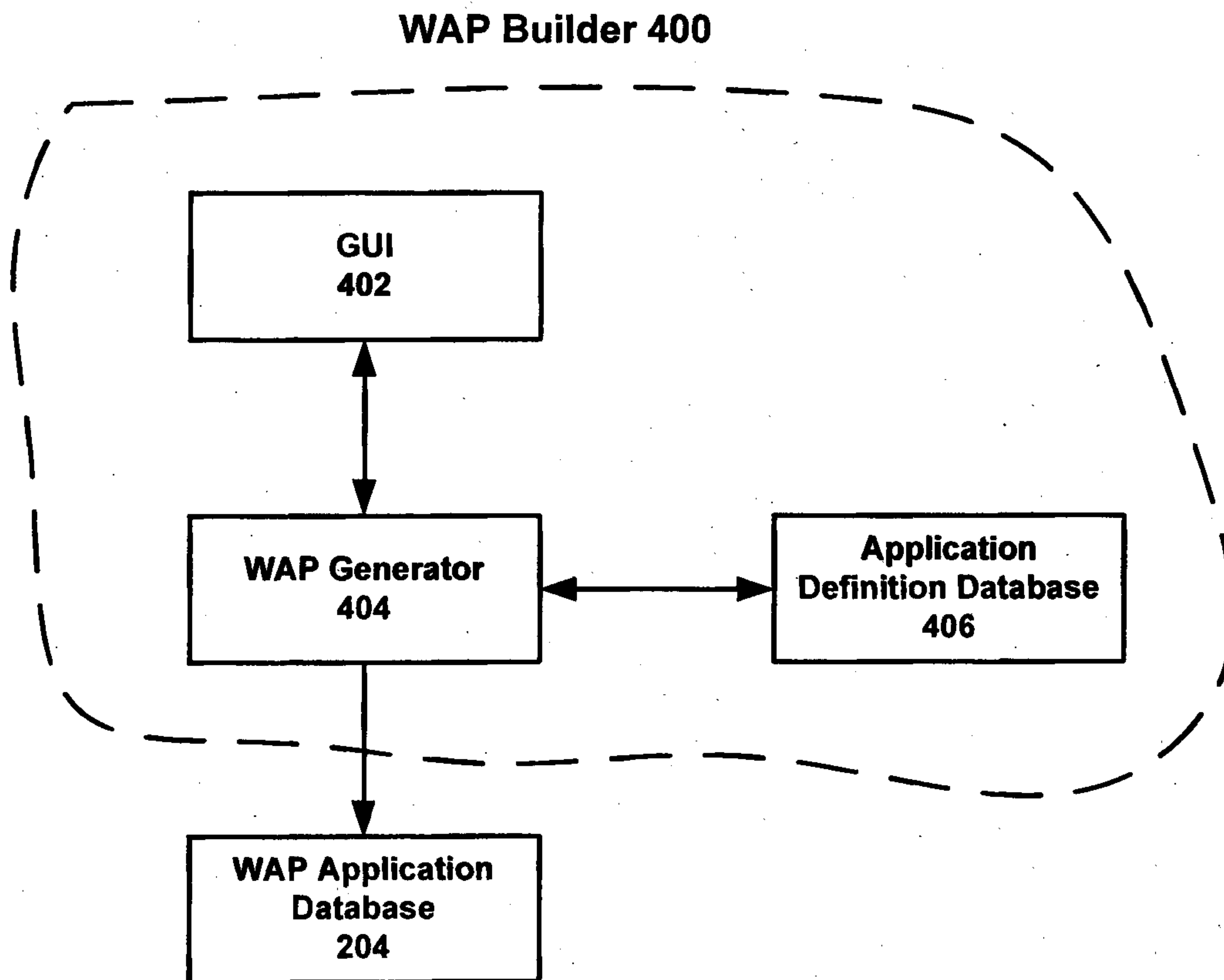


Fig. 4

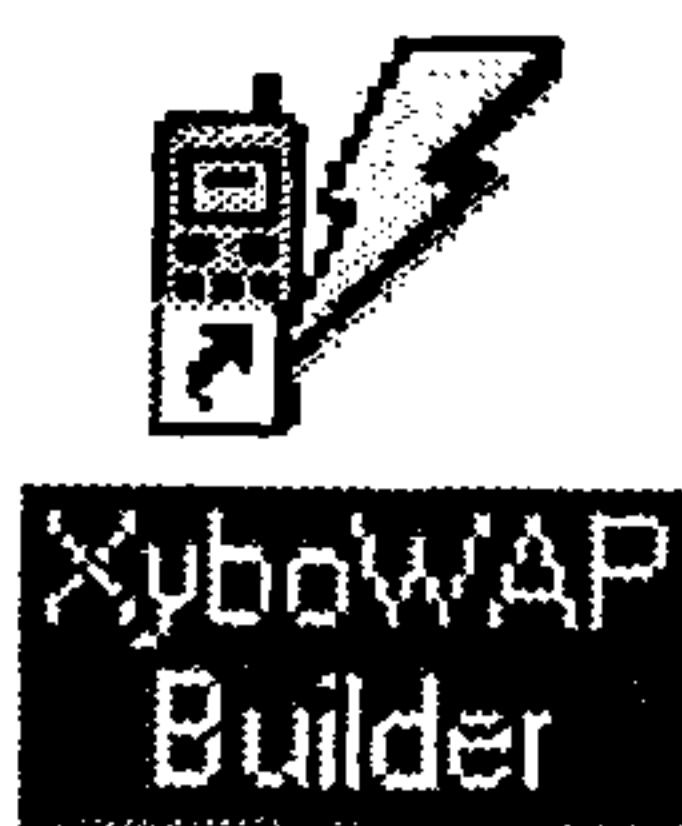


Figure 5a

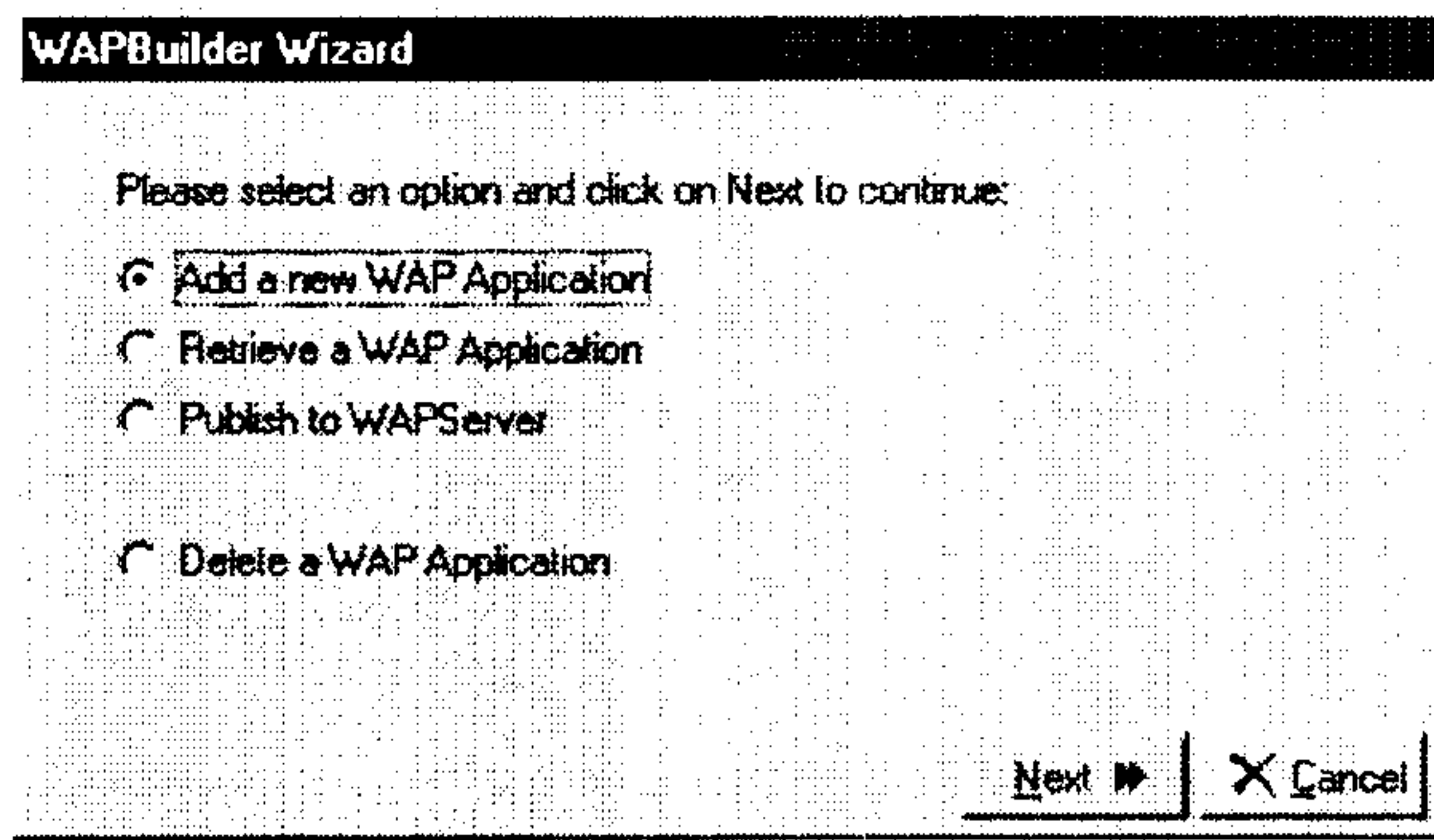


Figure 5b

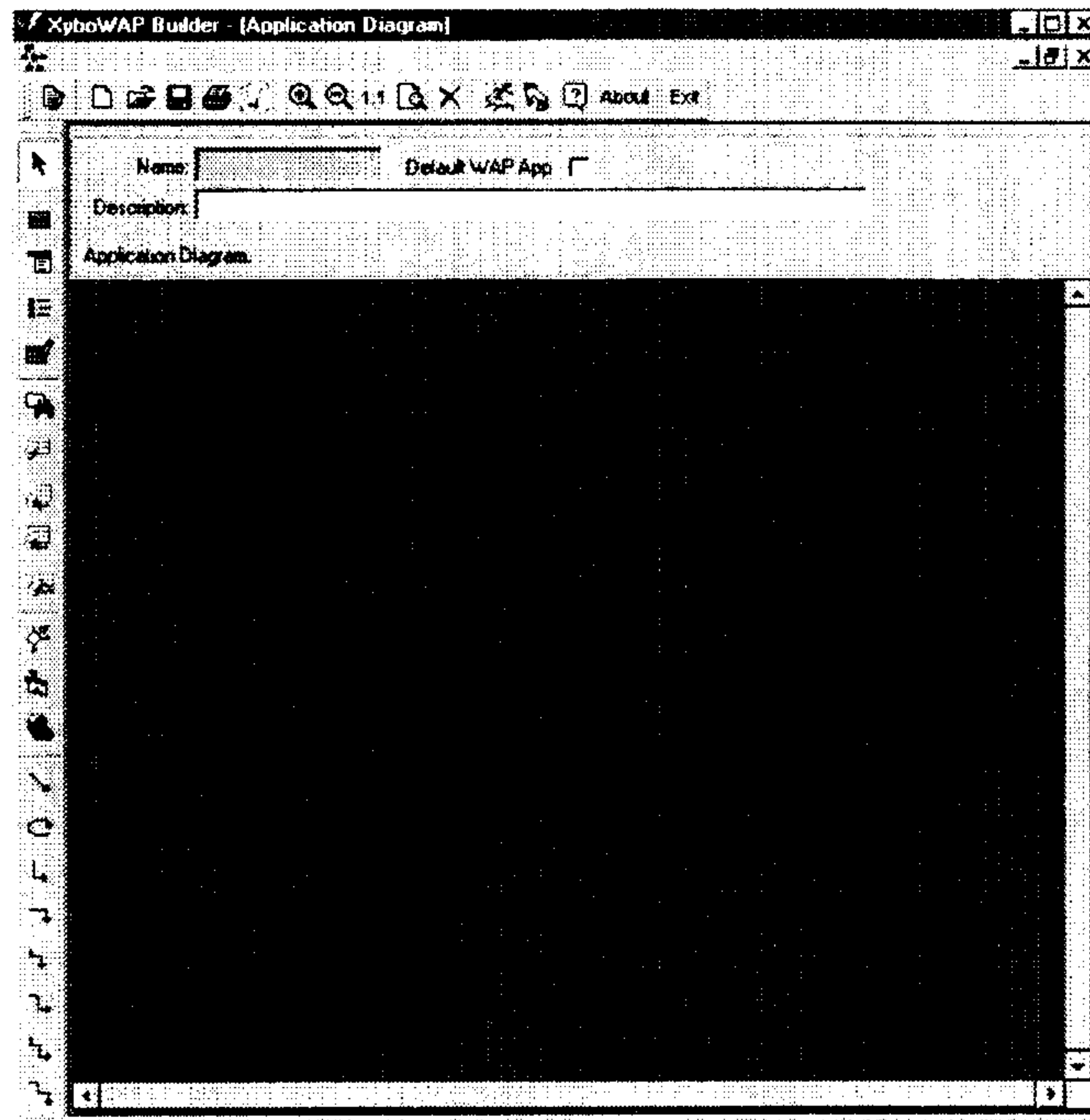


Figure 5c

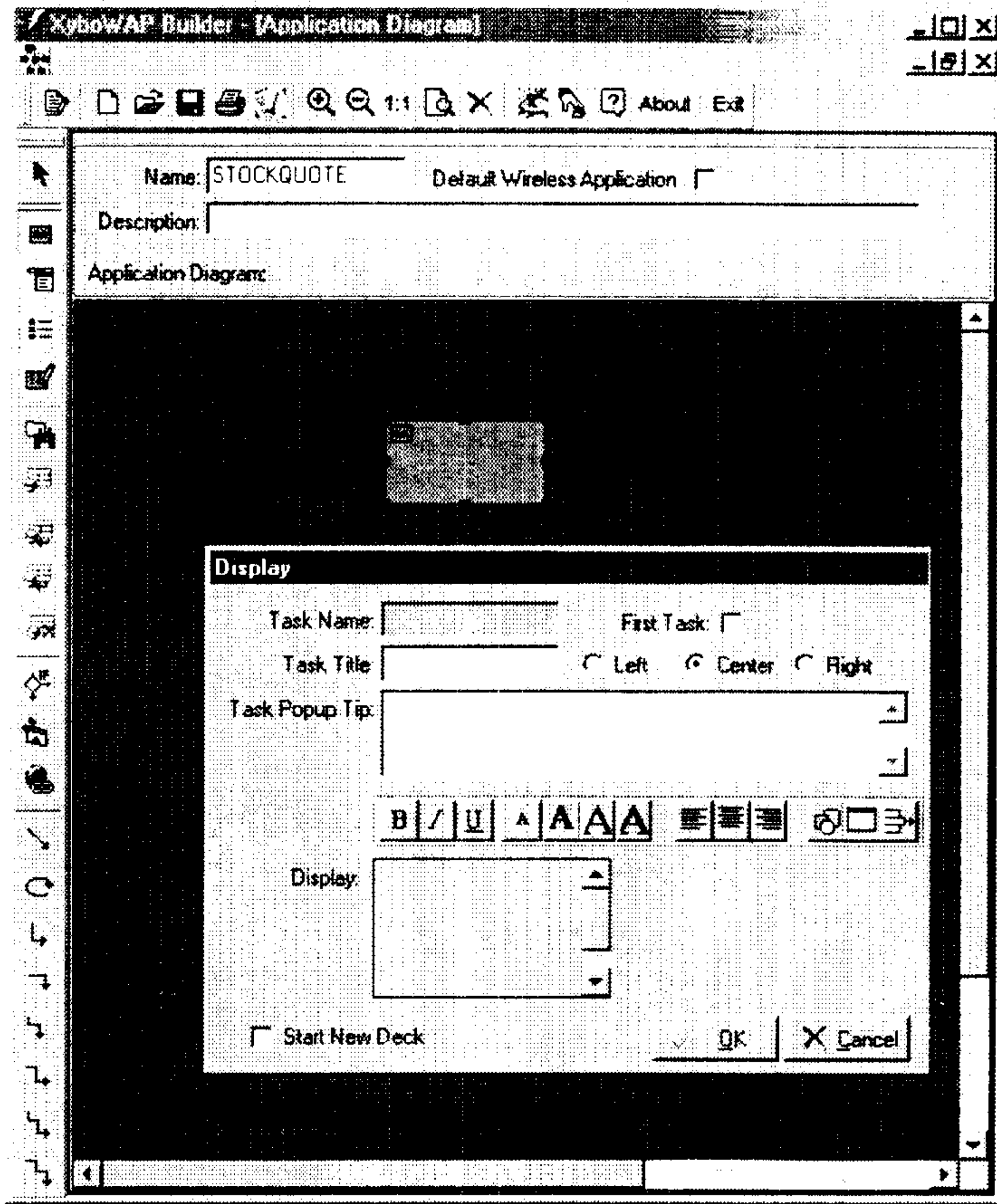


Figure 5d

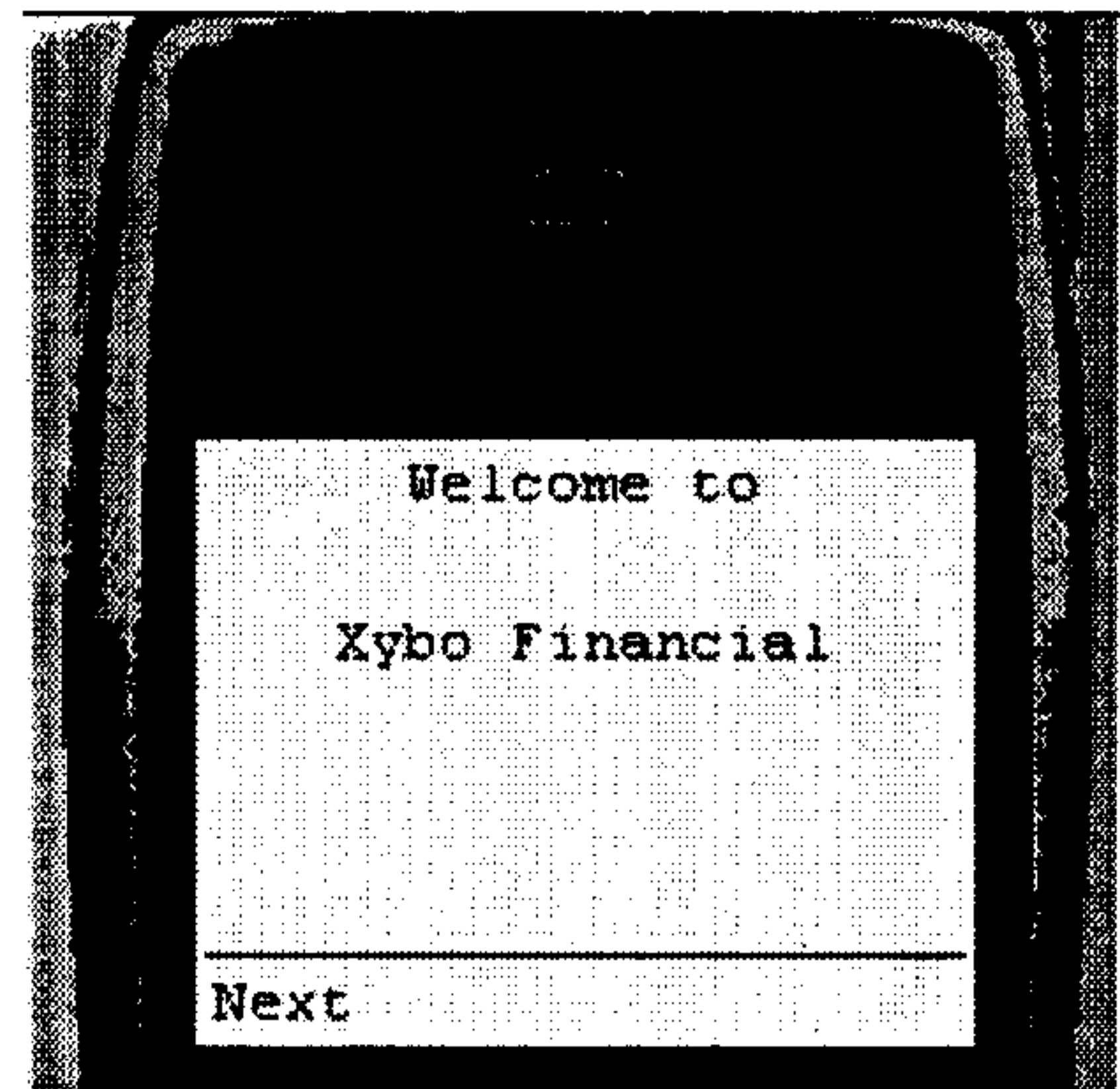
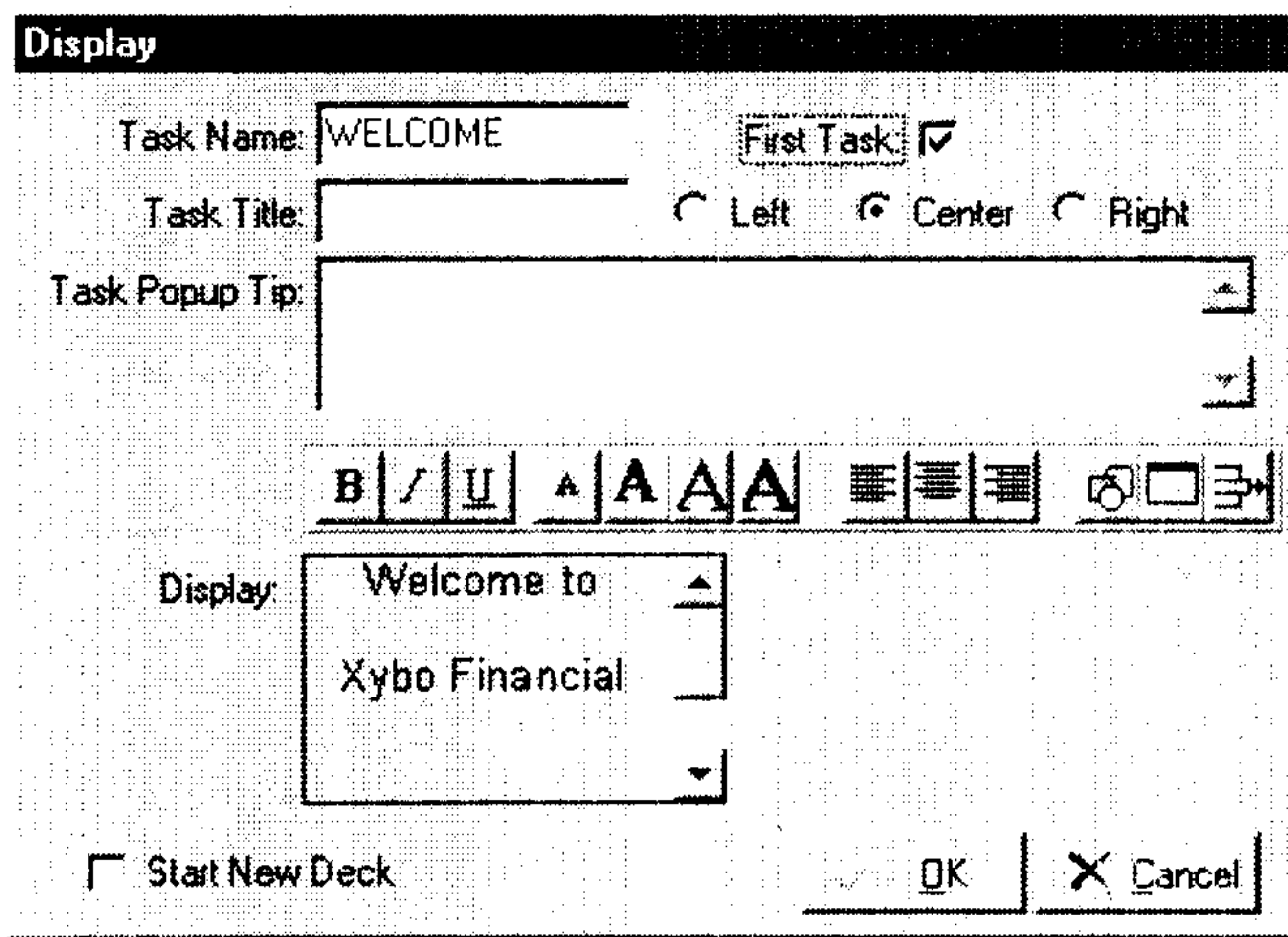


Figure 5e

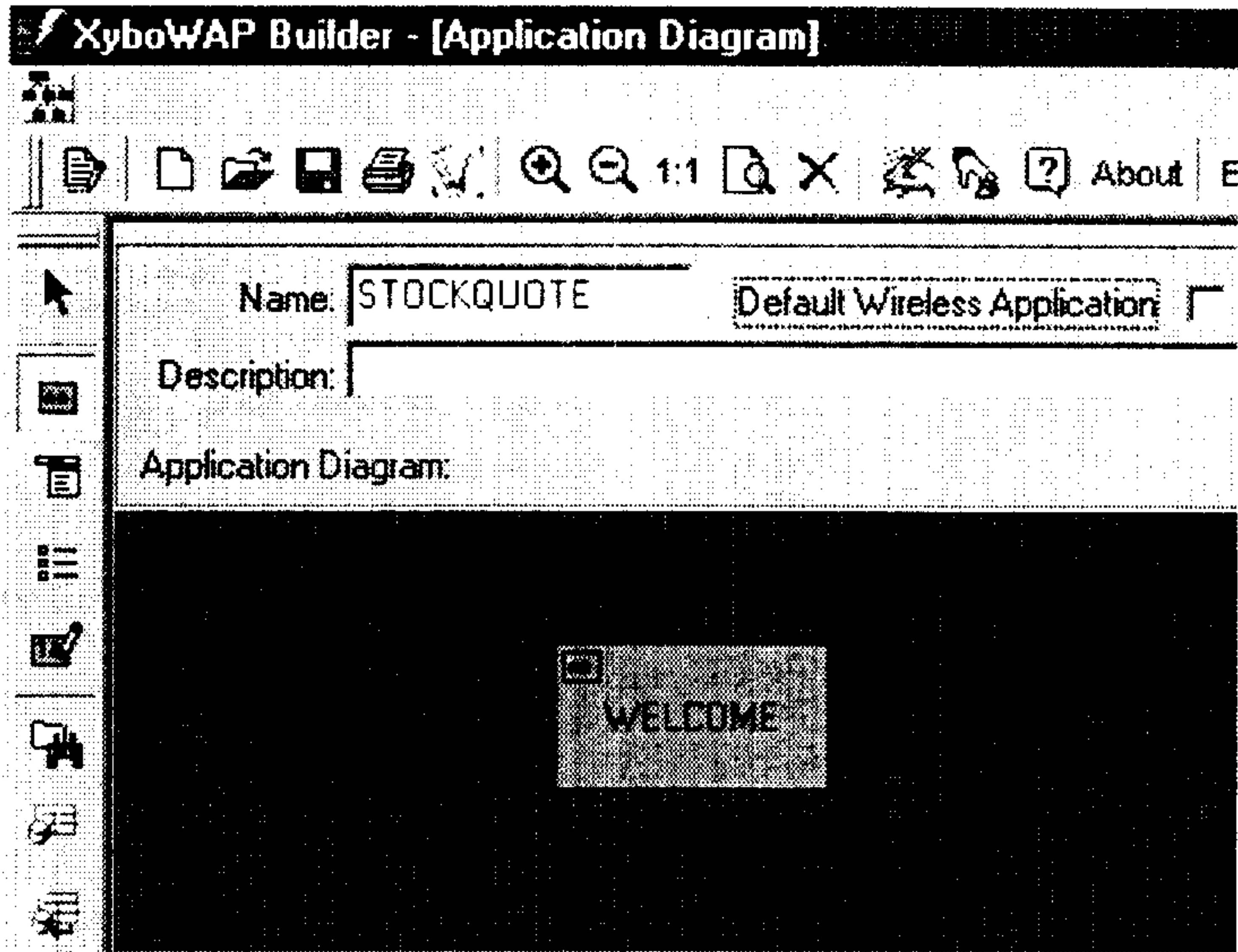


Figure 5f

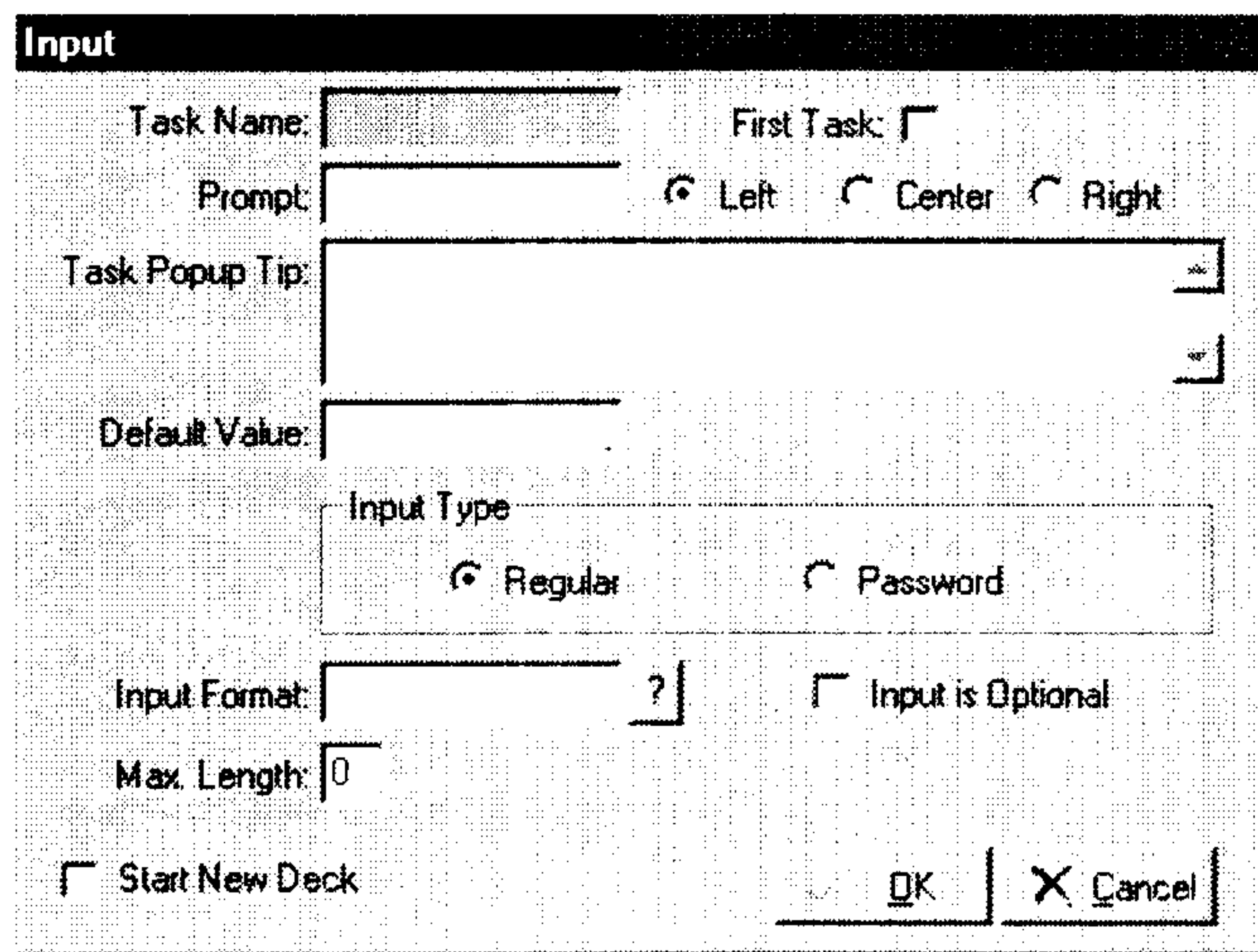


Figure 5g

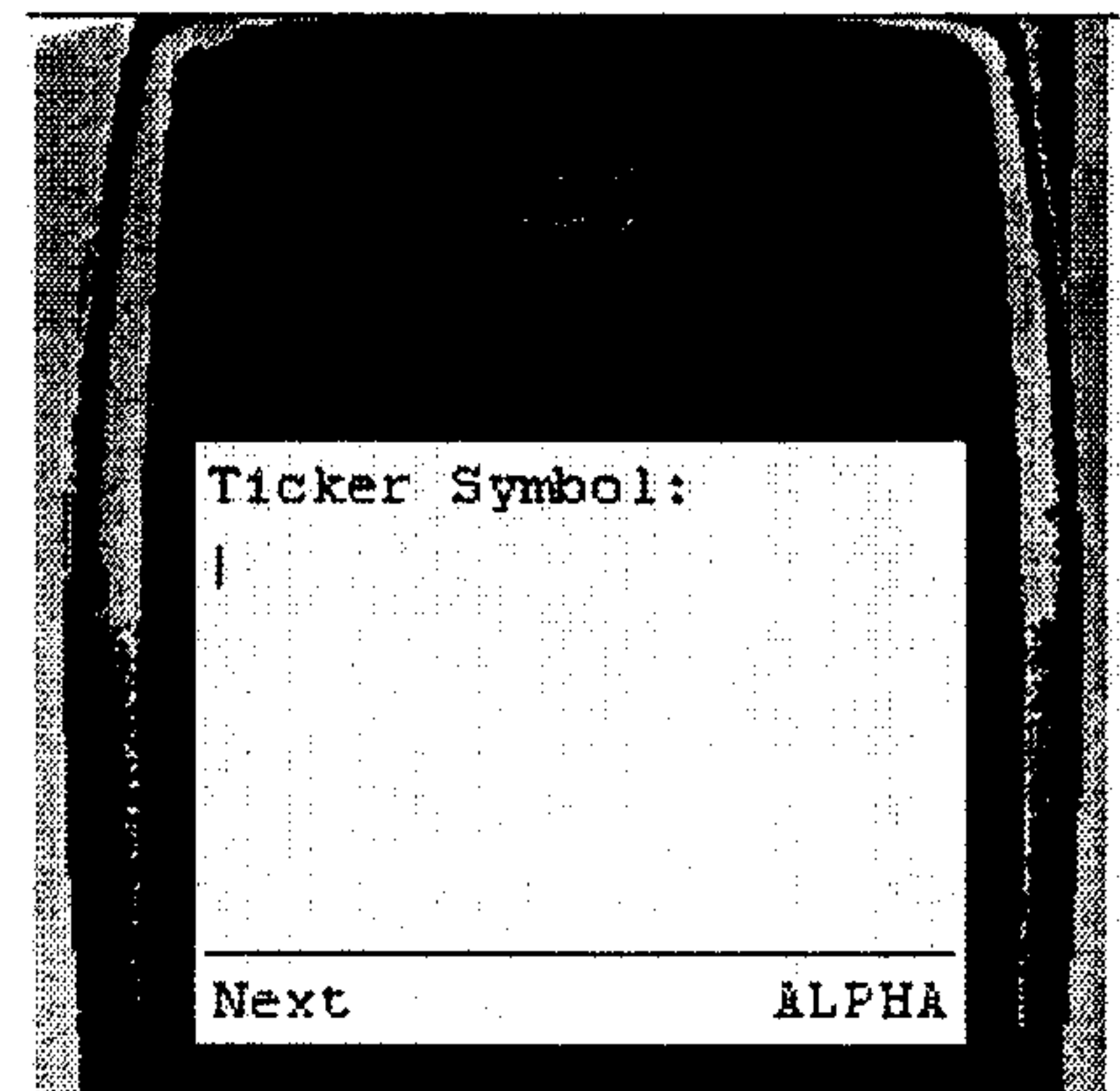
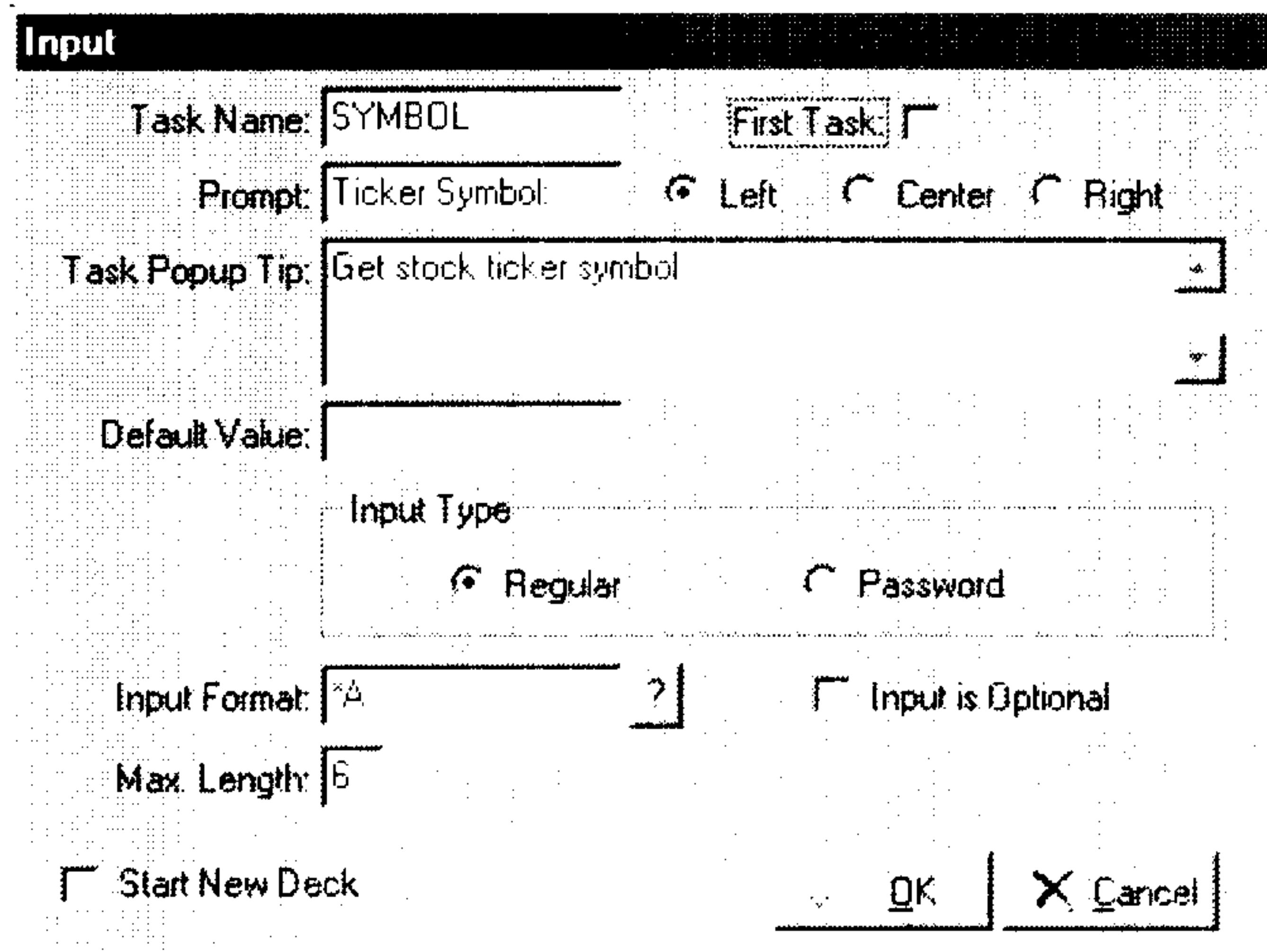


Figure 5h

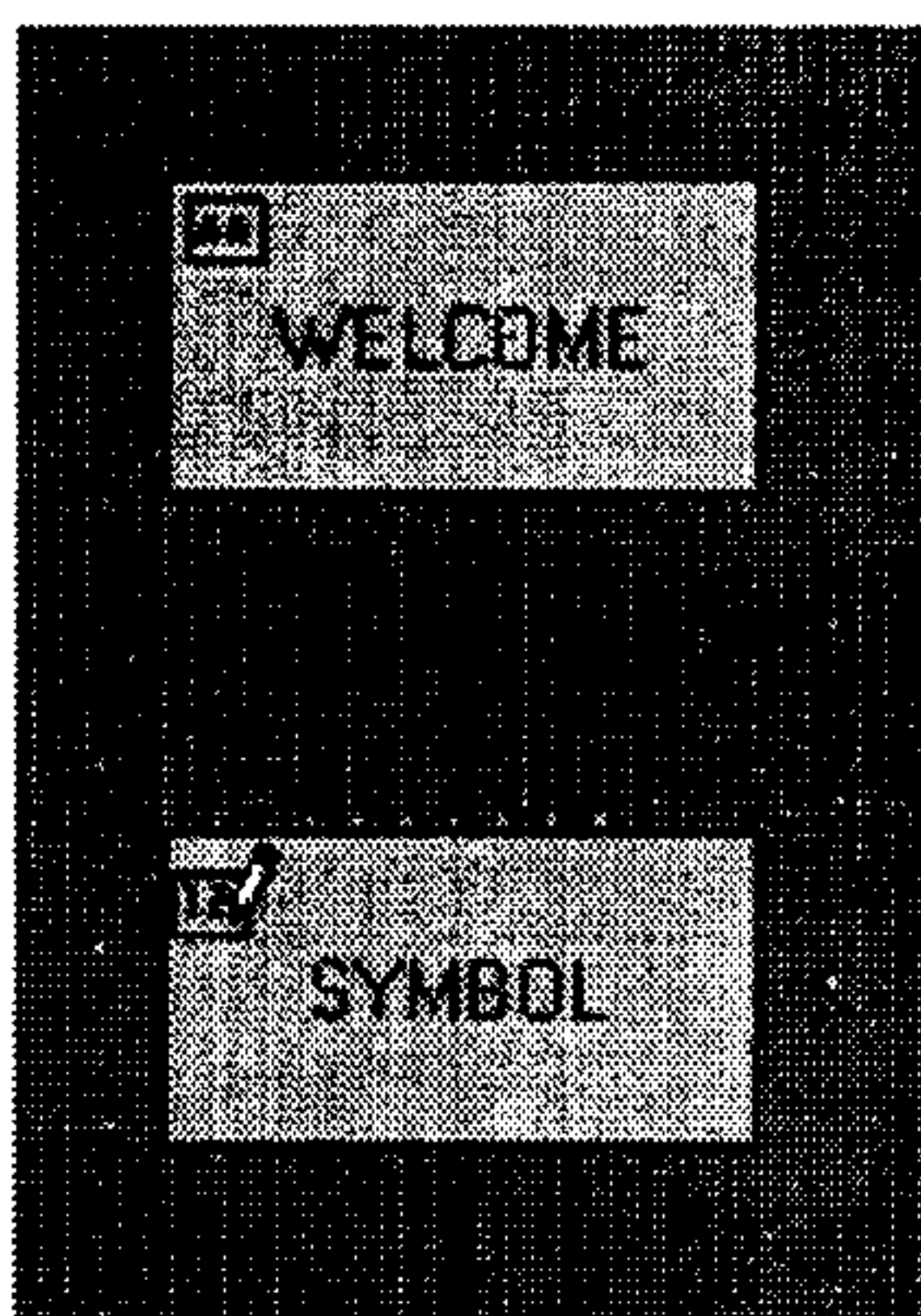


Figure 5i

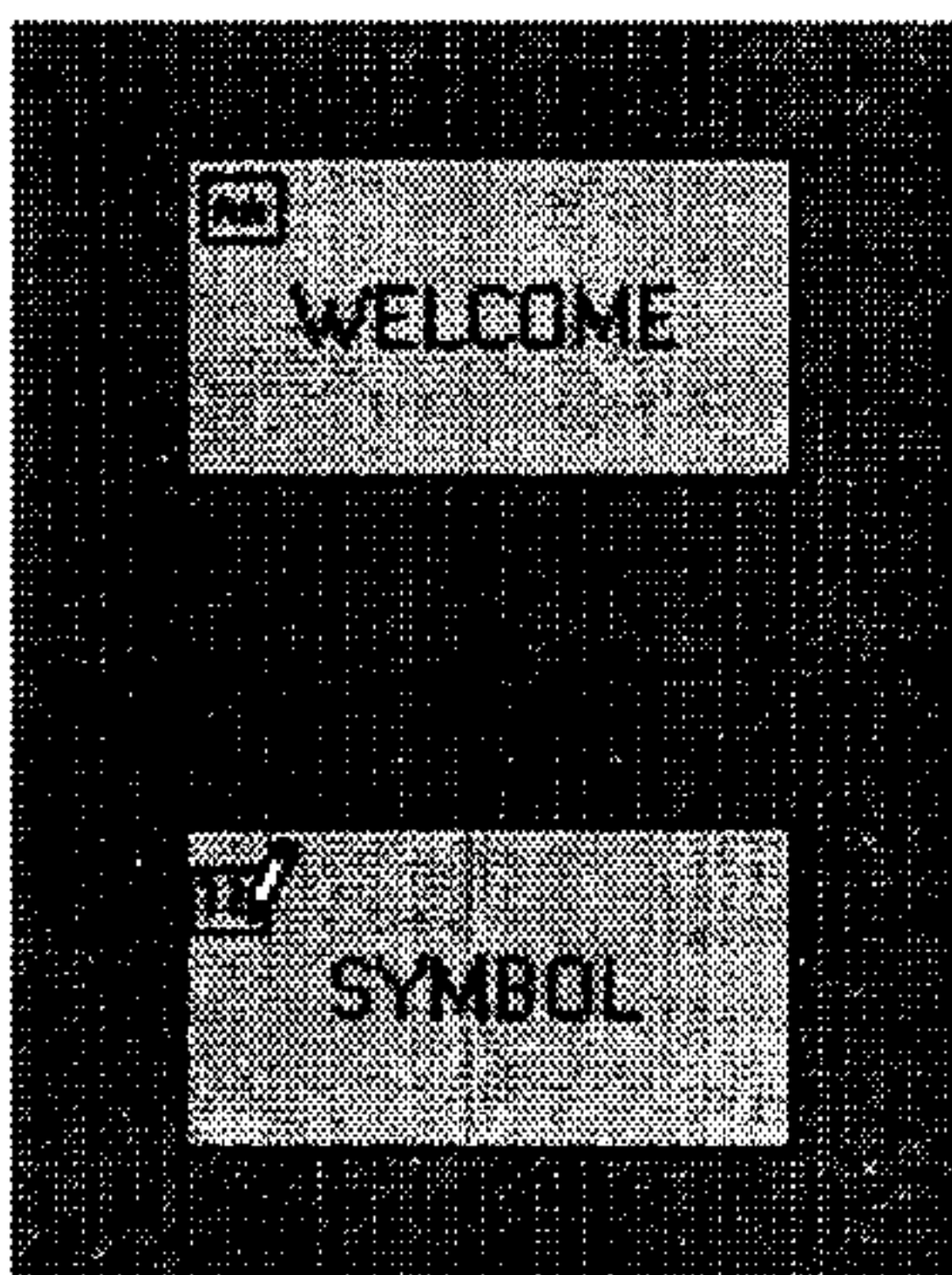


Figure 5j

Display Menu

Task Name: MAINMENU First Task:

Task Title: Left Center Right

Task Popup Tip: Get menu choice

 Menu Alignment: Left Center Right

Hint	Display Text	Image	Image Alt. Text

Start New Deck

Figure 5k

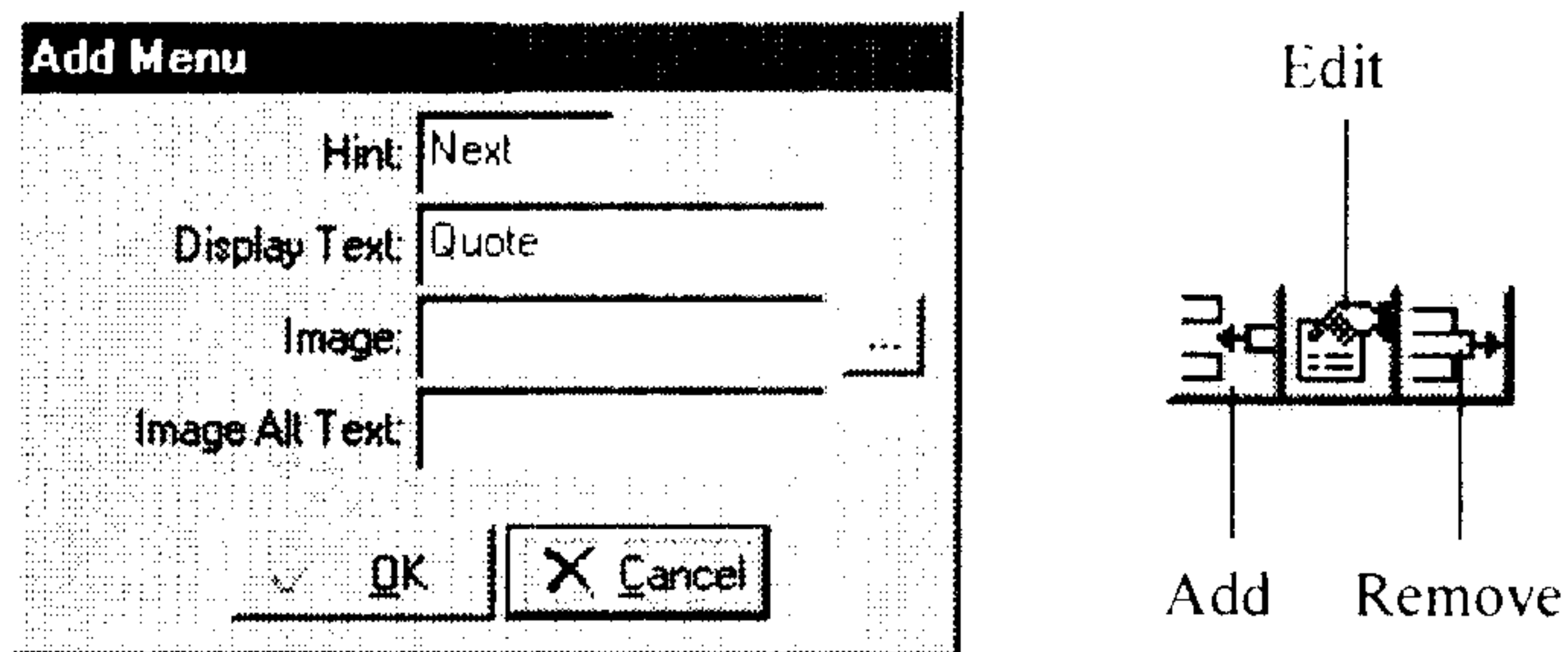


Figure 5l

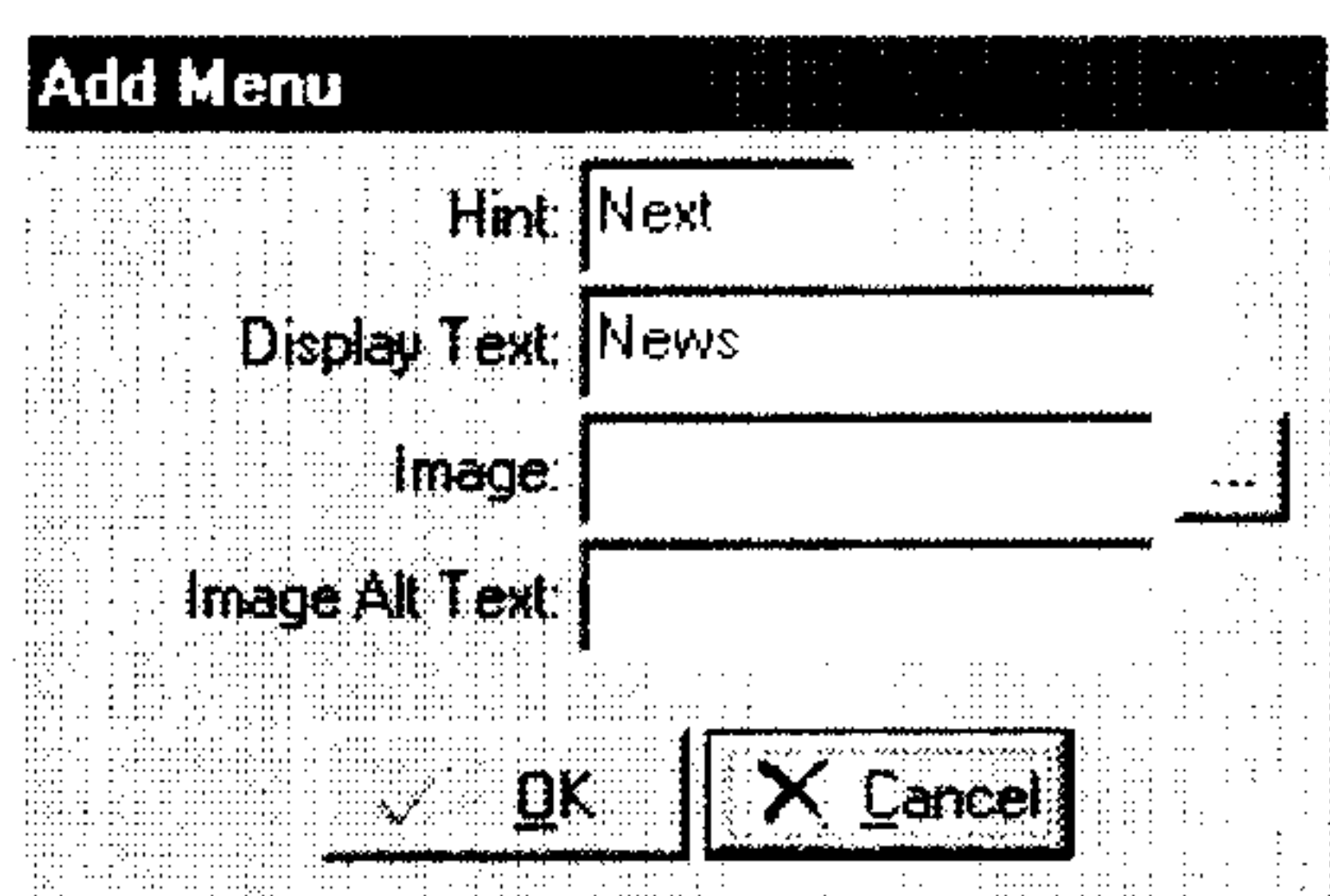


Figure 5m

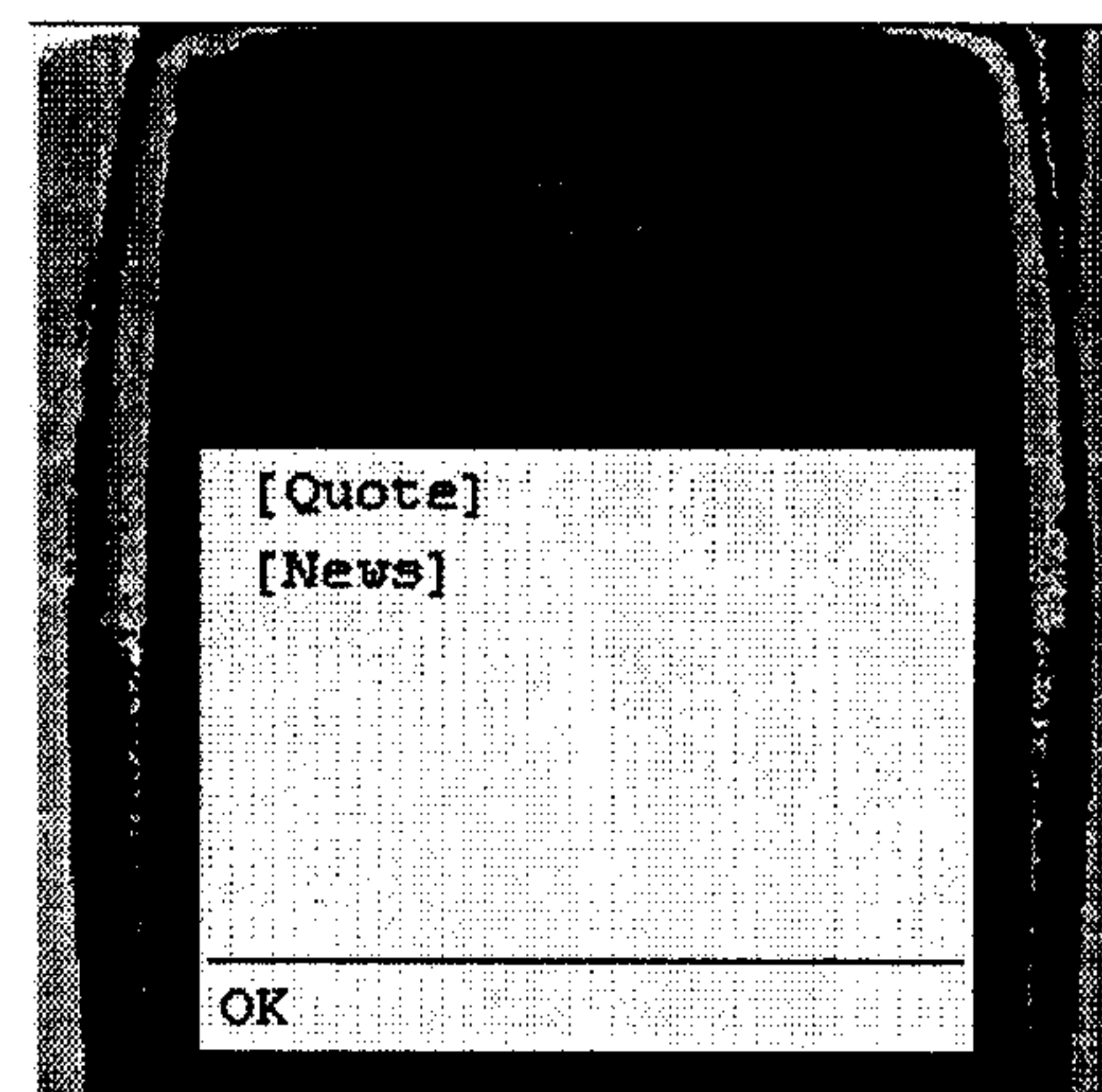
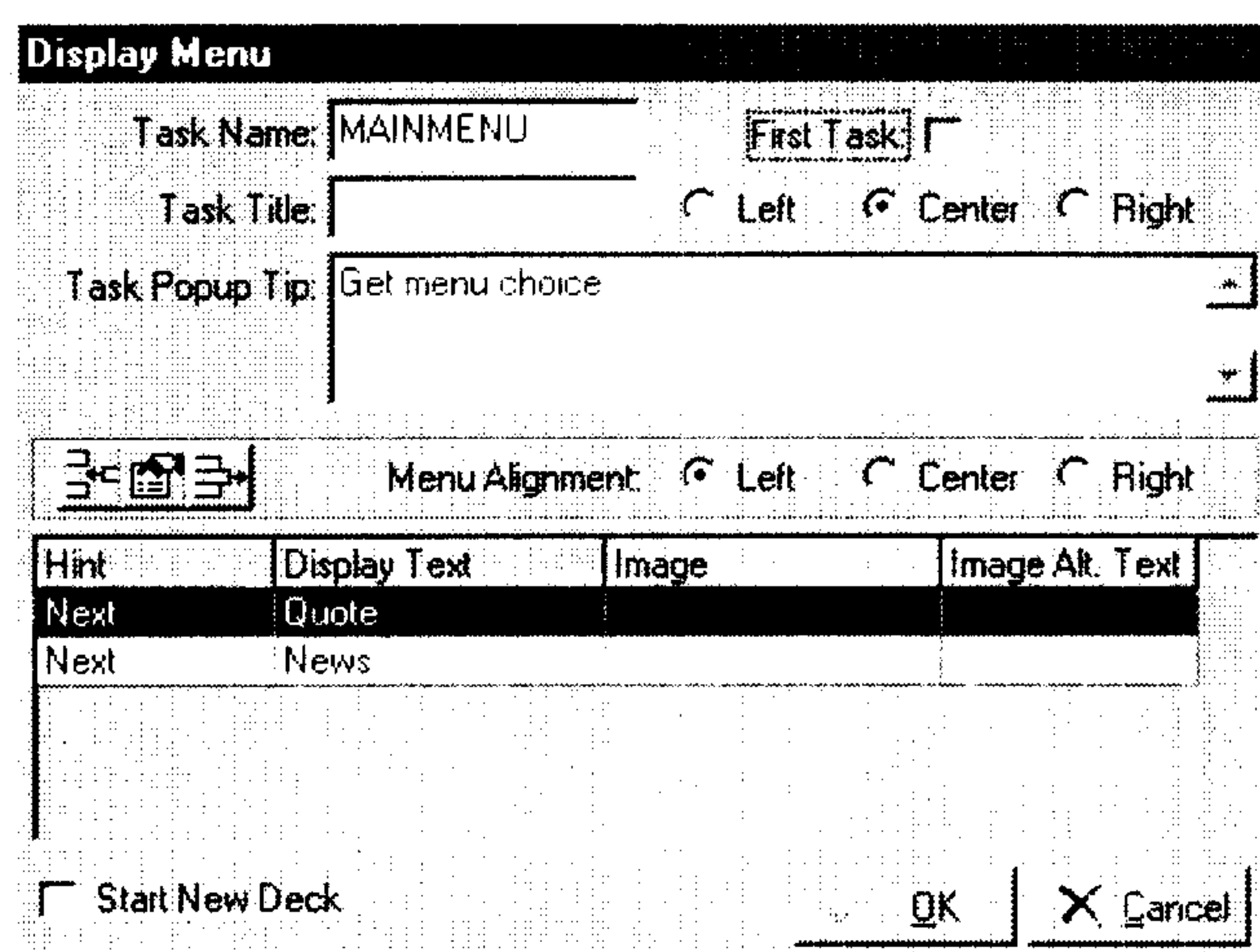


Figure 5n

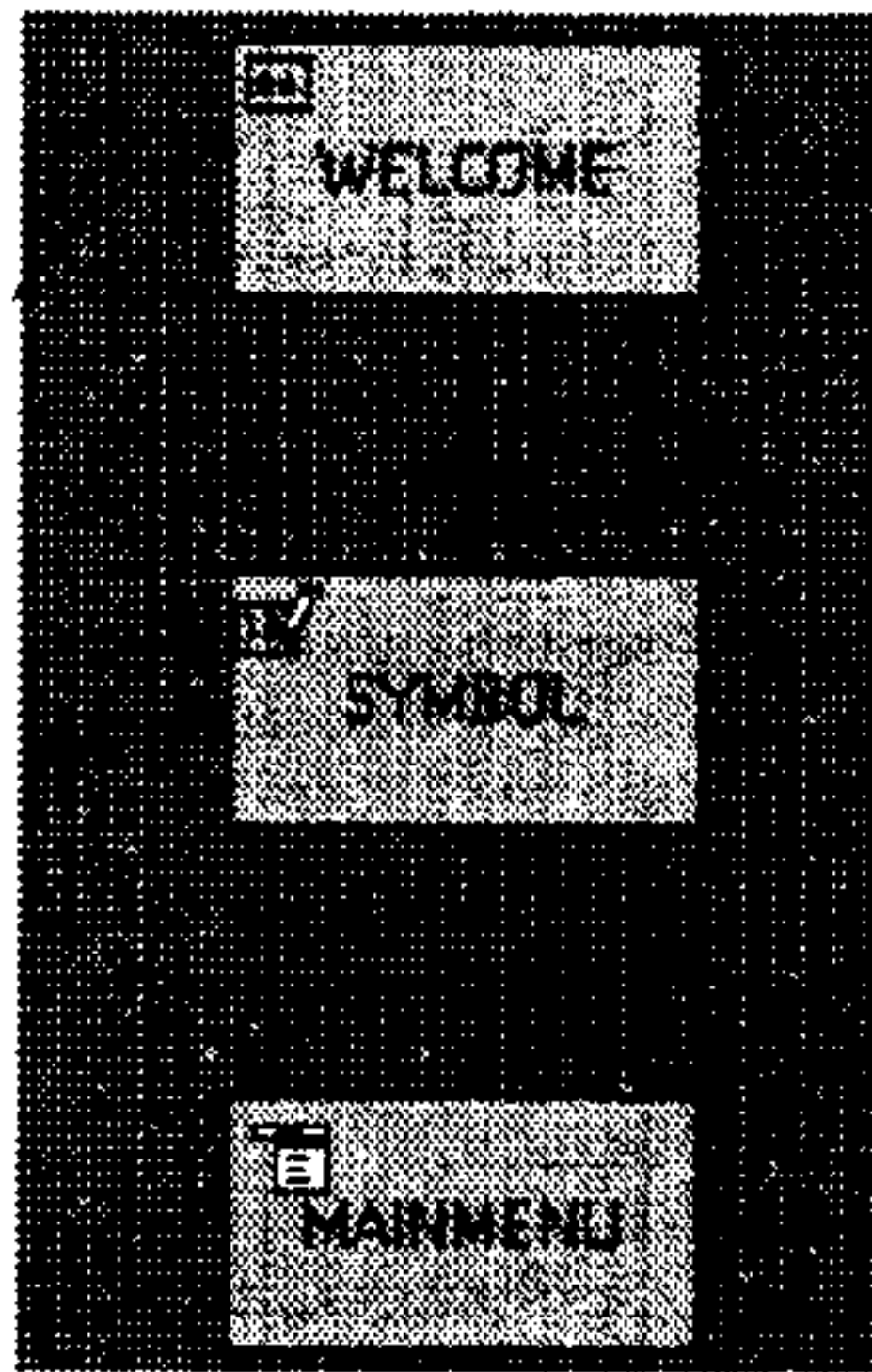


Figure 5o

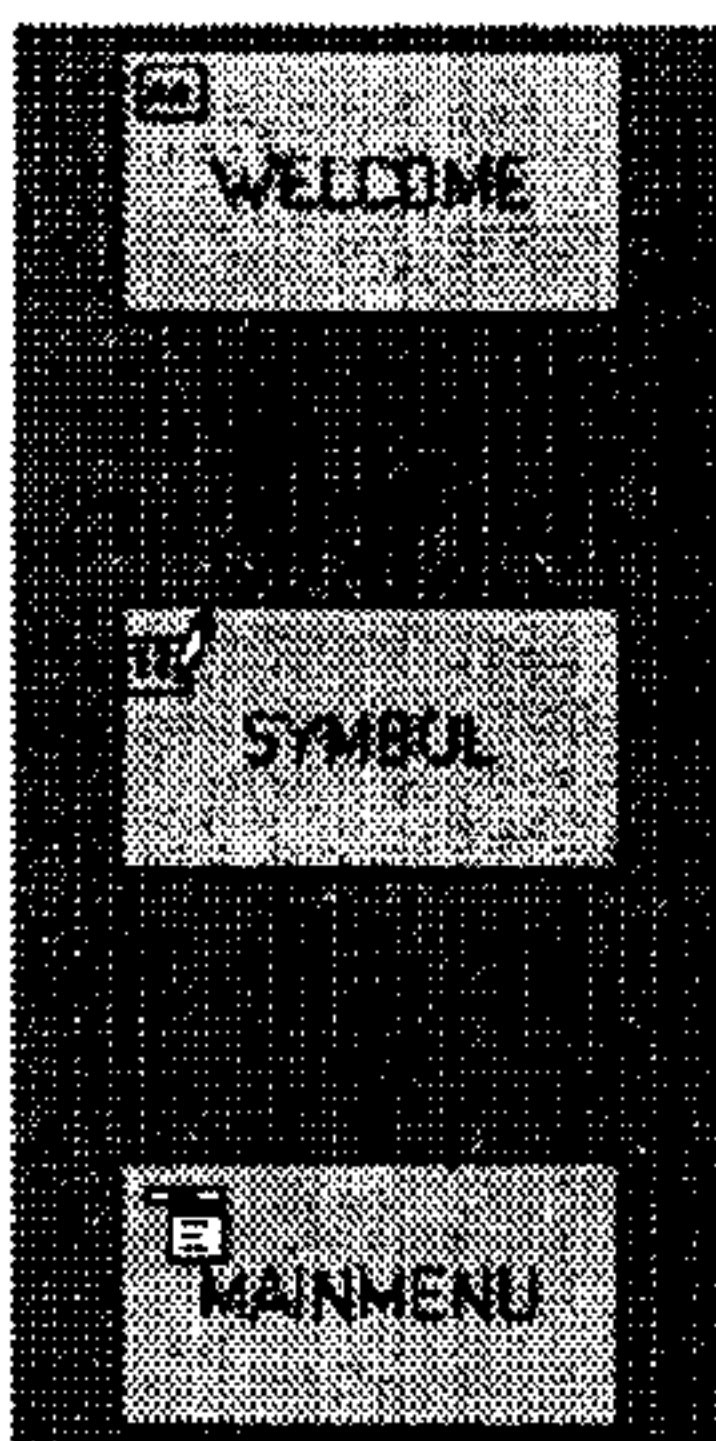


Figure 5p

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Data Source Login:

Data Source Password:

Data Source Name:

Table Name:

Return Action

Display Records Display as Options Display as Table

Return Record as Variable

Figure 5q

Data Retrieval - Display Data

Task Name: QUOTE First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: 0 Data Source Encoding Format: ASCII

Basic | Fields | Advance | Messages

Data Source Login:

Data Source Password:

Data Source Name:

Table Name: QUOTE

Return Action

Display Records Display as Options Display as Table

Return Record as Variable

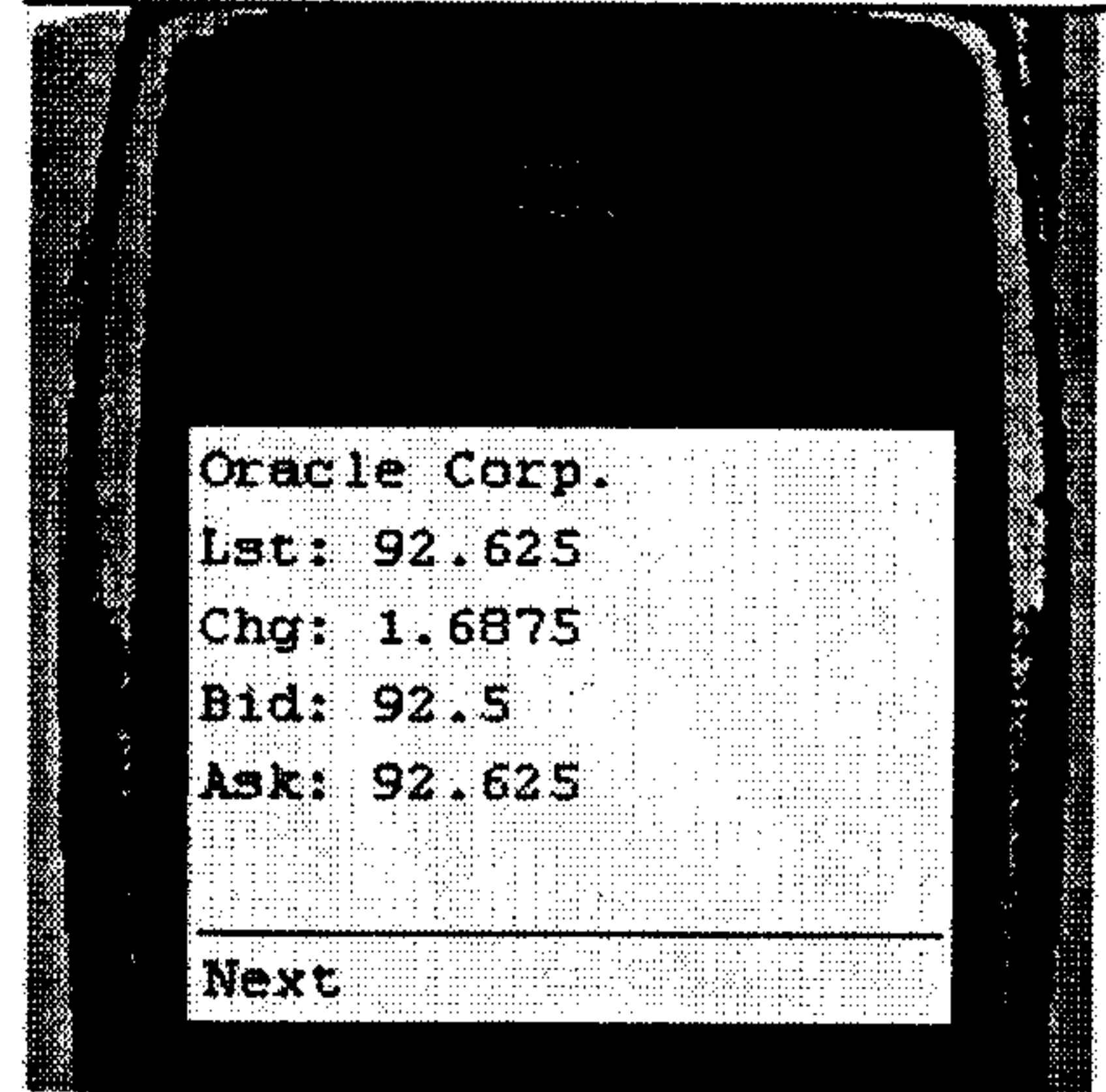


Figure 5r

Data Retrieval - Display Data

Task Name: QUOTE First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: 0 Data Source Encoding Format: ASCII

Basic | Fields | Advance | Messages

Field Name	Header	Type	Visible

Criteria:

Order by:

Return Record as Variable

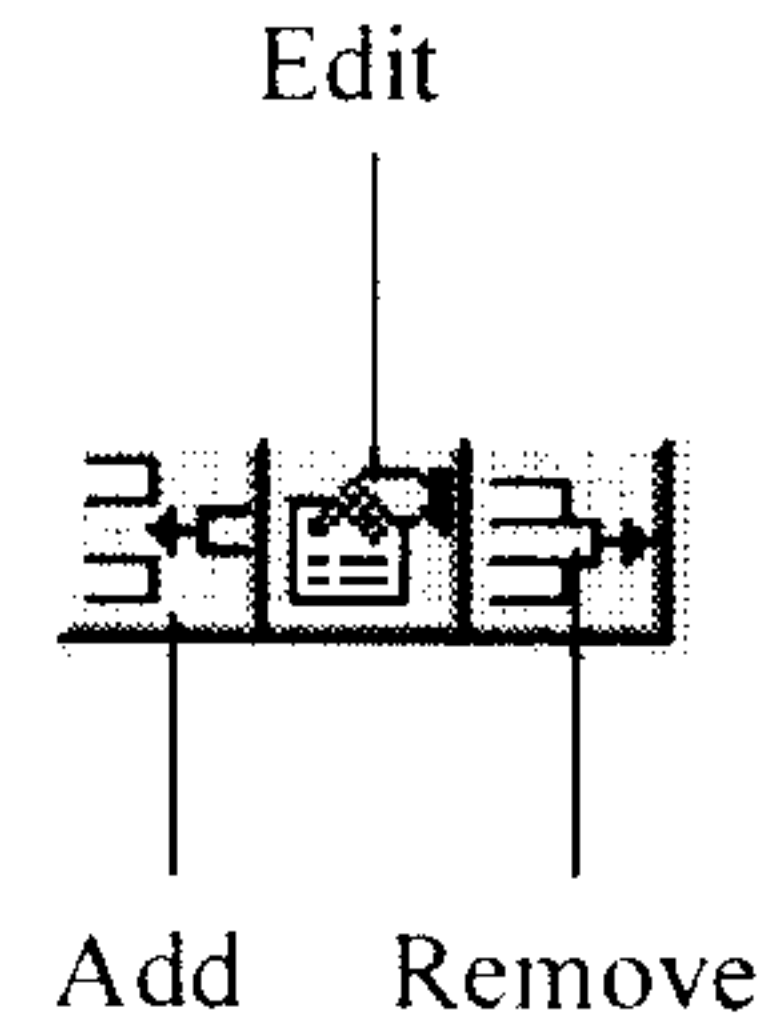


Figure 5s

Add Field Information

Field Name:

Header:

Type: Alpha Numeric

Visible

Figure 5t

Add Field Information

Field Name:

Header:

Type: Alpha Numeric

Visible

Figure 5u

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Field Name	Header	Type	Visible
Name		Alpha	Yes
Last	Lst. :	Numeric	Yes
Change	Chg. :	Numeric	Yes

Criteria:

Order by:

Return Record as Variable

Figure 5v

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

ADO Connection String:

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=\$YBOPATH\MDB\Stock.mdb

SQL String:

Return Record as Variable

Figure 5w

Data Retrieval - Display Data

Task Name: QUOTE First Task:

Task Title: Left Center Right

Task Popup Tip: []

Max. Records Return: 0 Data Source Encoding Format: ASCII

Basic | Fields | Advance | Messages

Operation Successful: []

Operation Failed: []

Failed to get quote

Error Encountered: []

Error getting quote Display Standard Message

Empty Record set: []

Invalid ticker symbol Display Standard Message

Return Record as Variable [OK] [Cancel]

Figure 5x

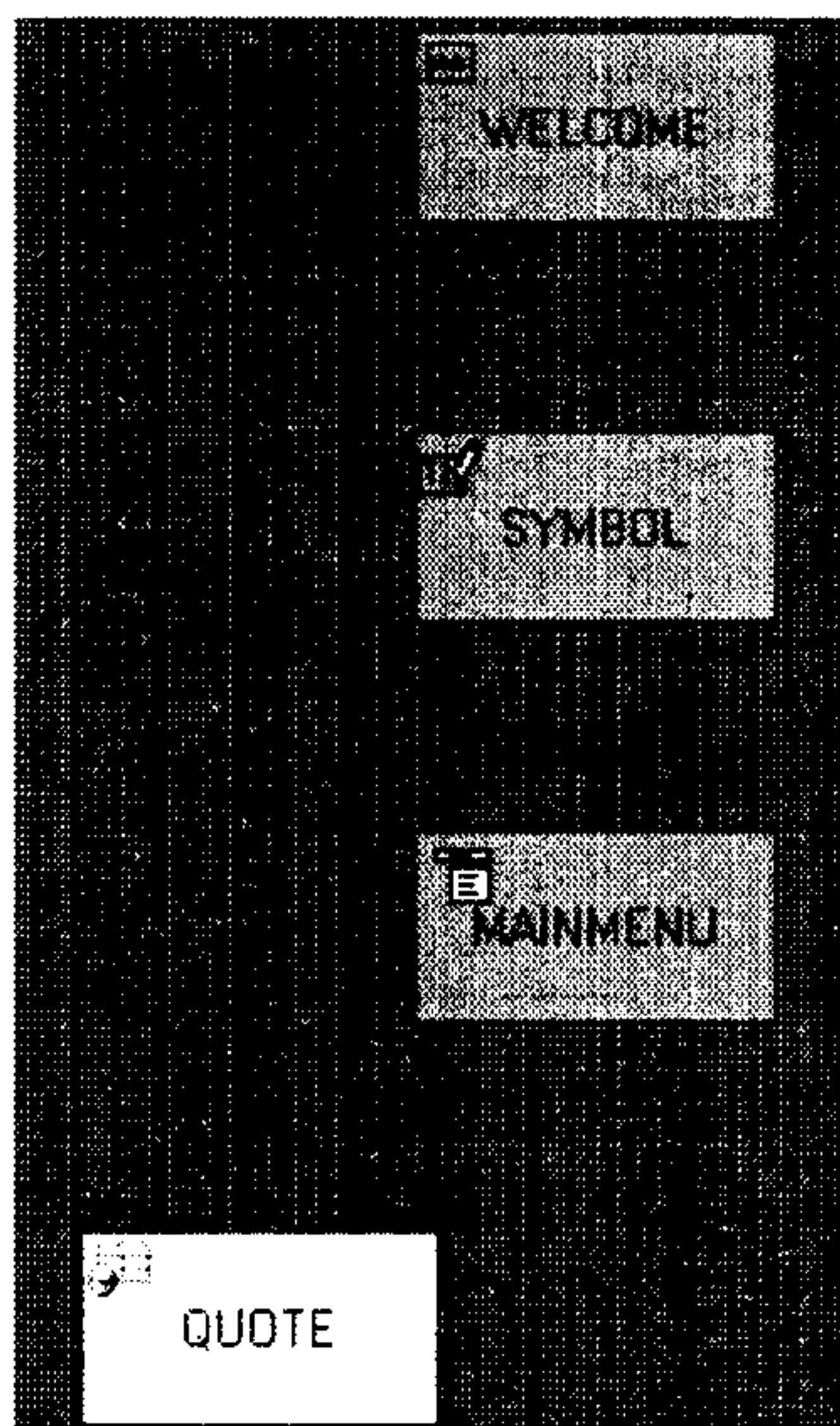


Figure 5y

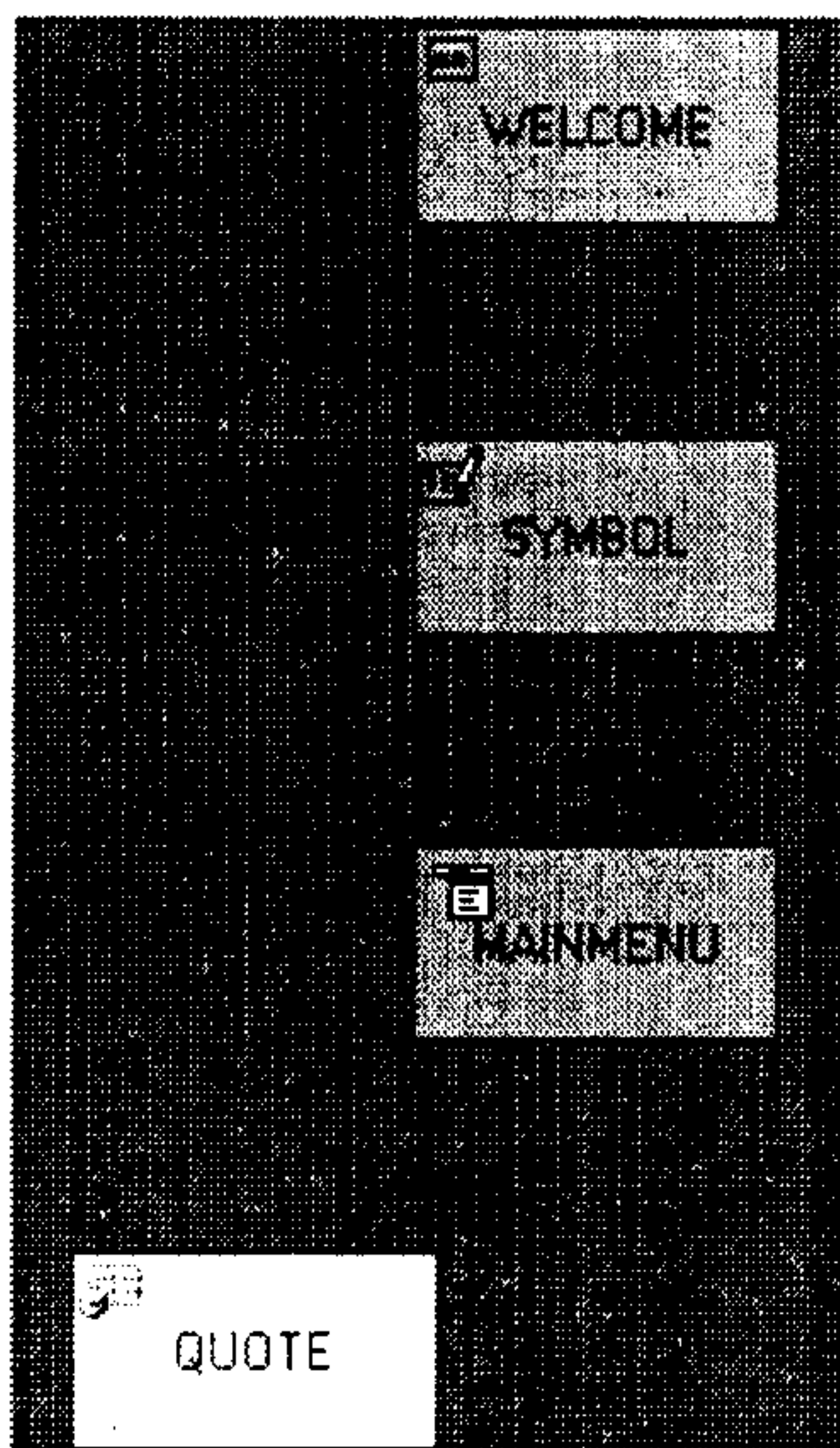


Figure 5z

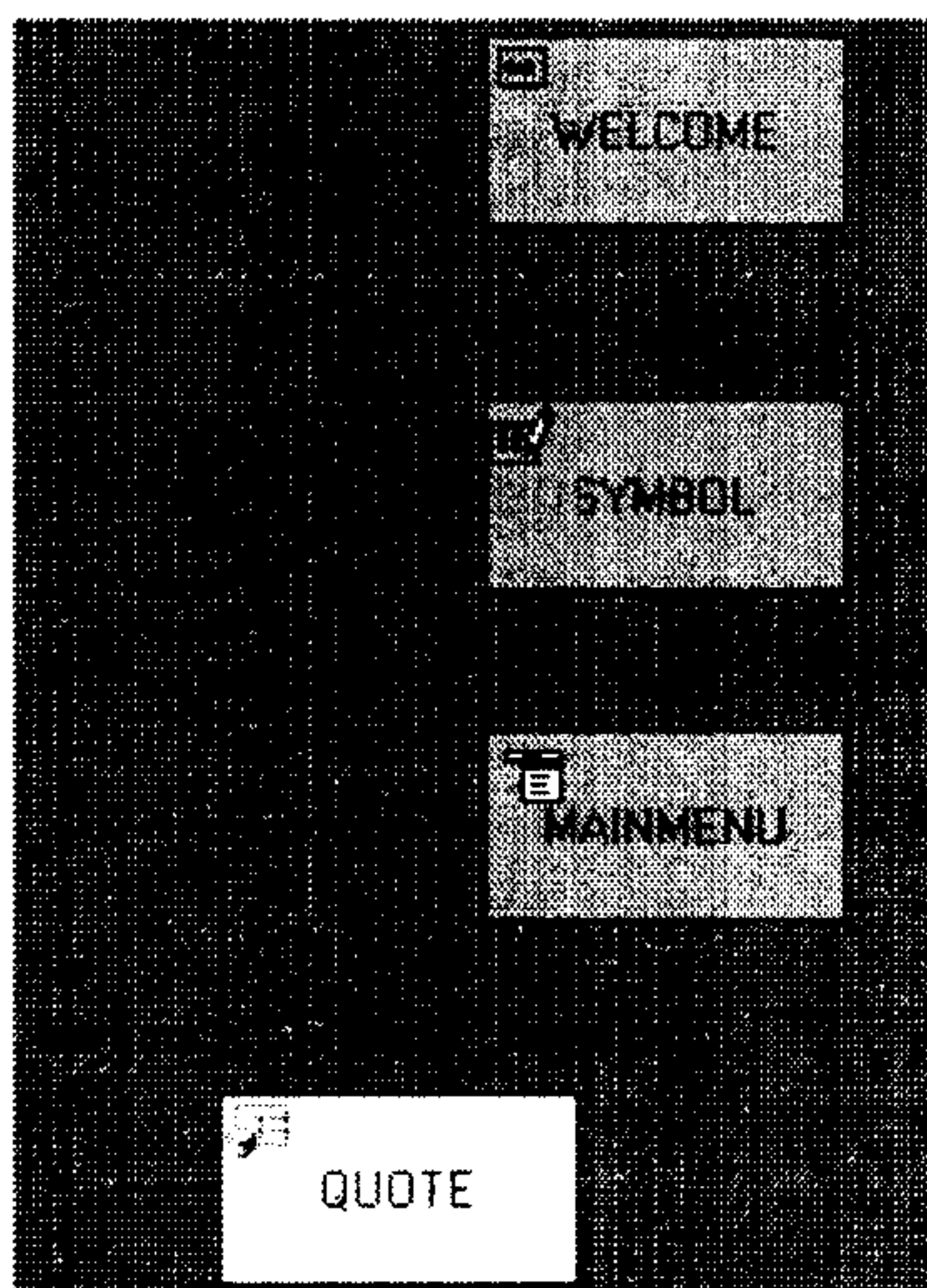


Figure 5aa

Display Menu

Task Name: MENU2 First Task:

Task Title: Left Center Right

Task Popup Tip: Prompt for detailed quote or next quote

 Menu Alignment: Left Center Right

Hint	Display Text	Image	Image Alt. Text

Start New Deck

Figure 5bb

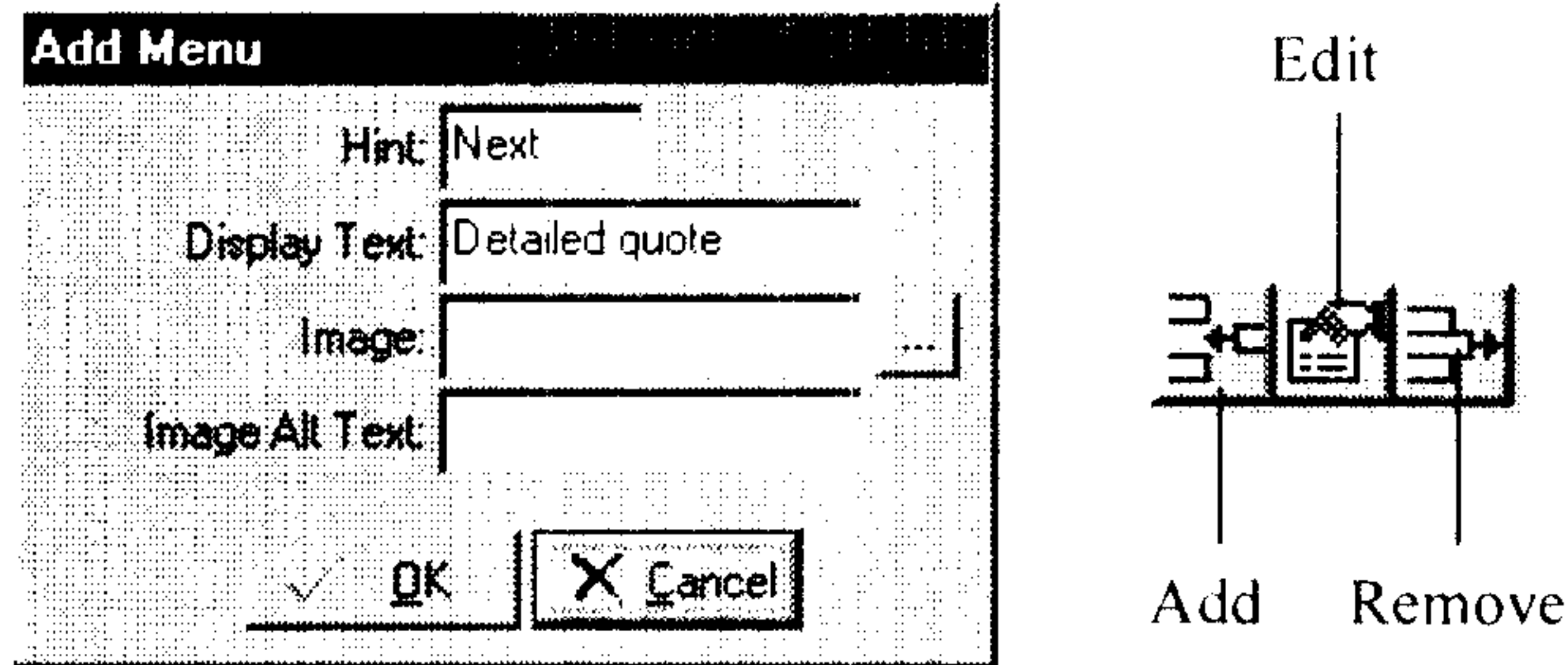


Figure 5cc

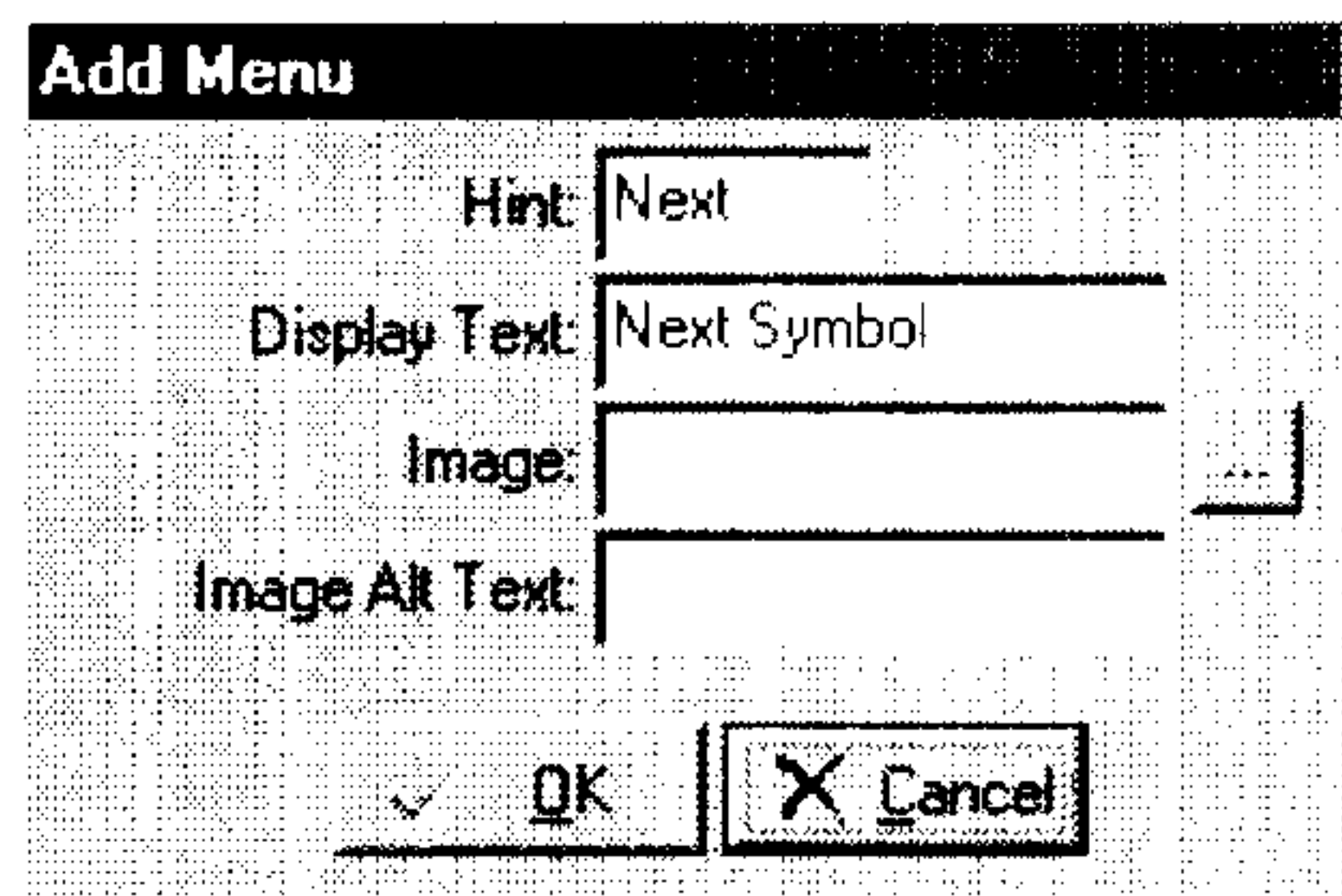


Figure 5dd

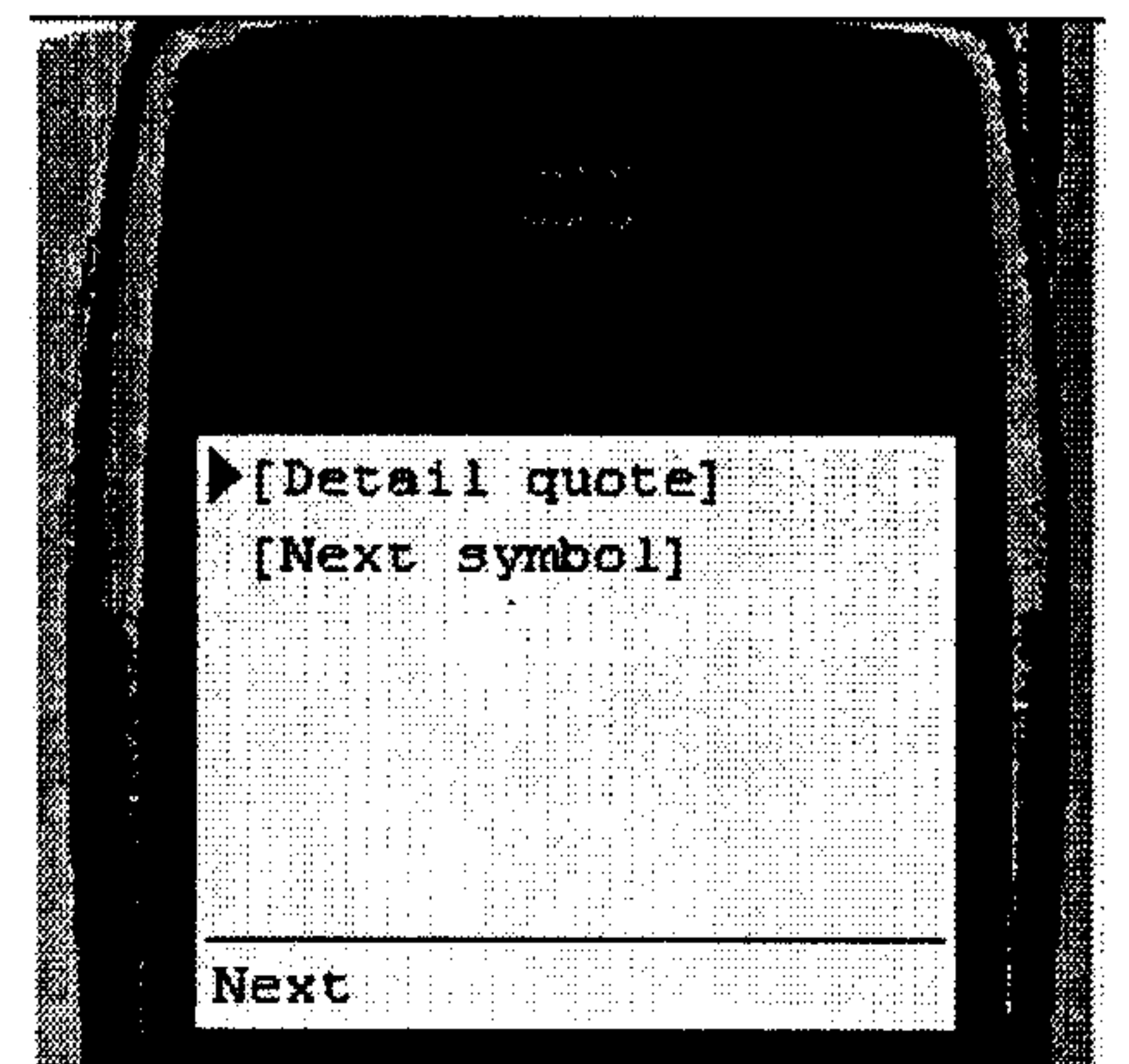
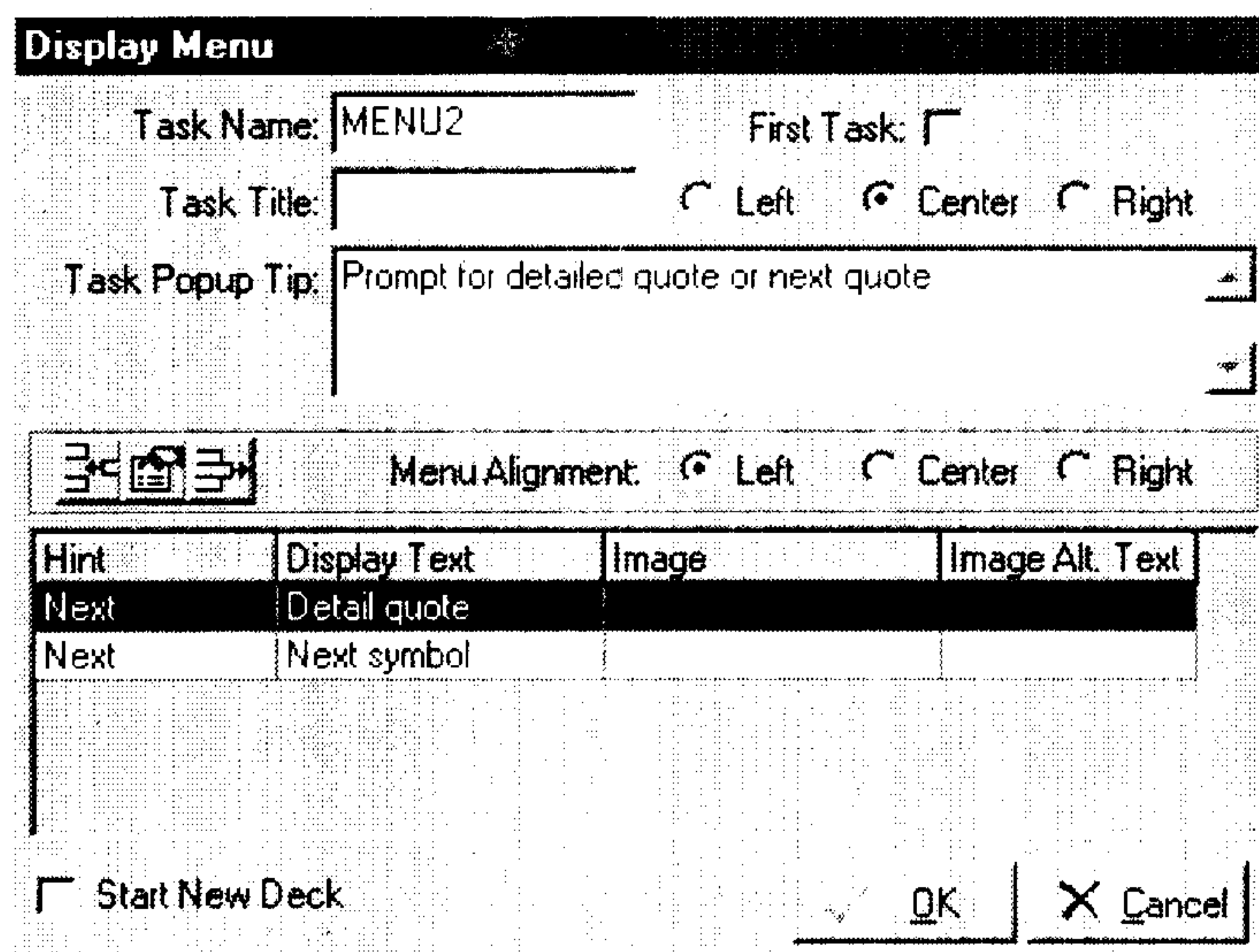


Figure 5ee

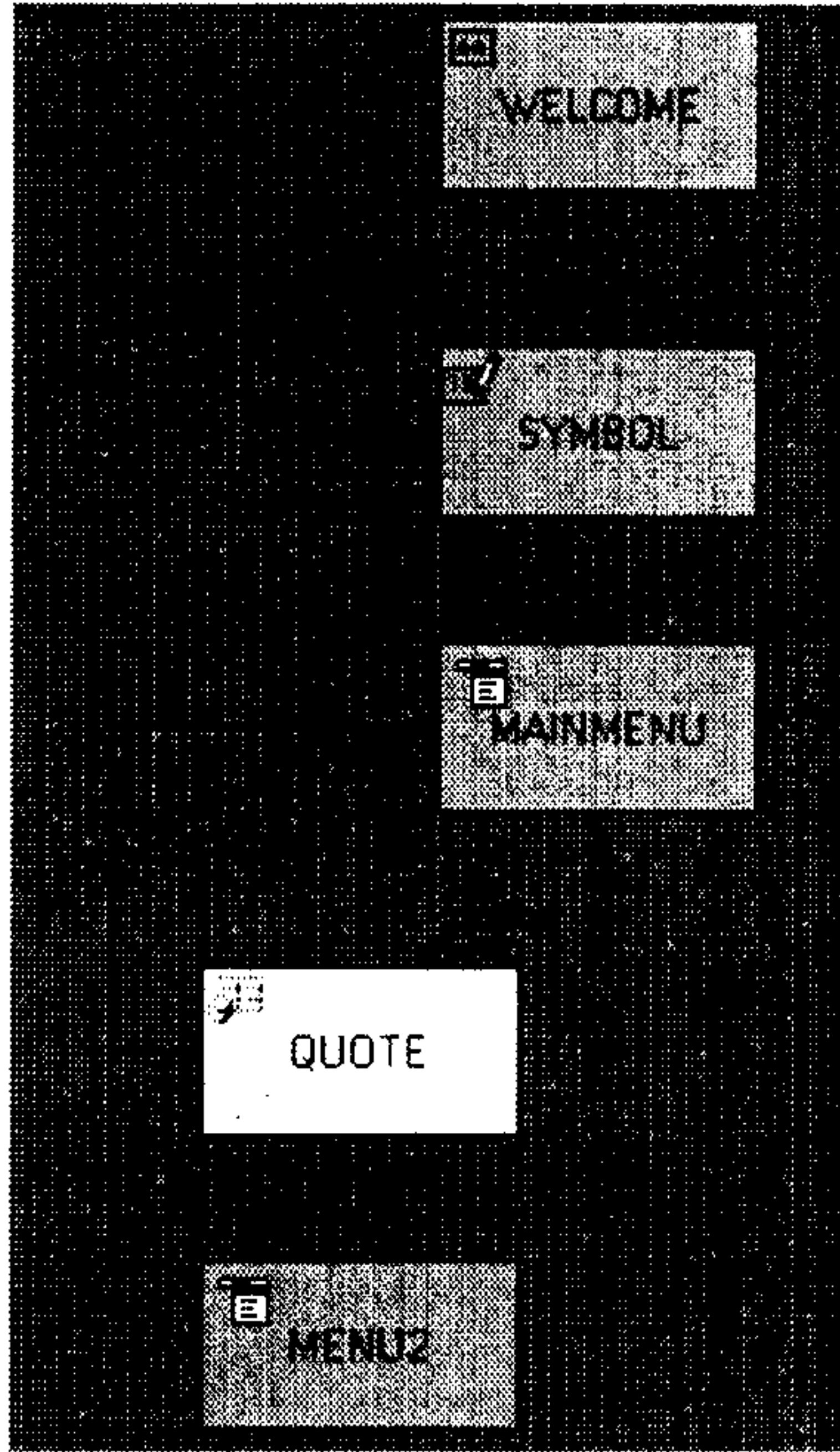


Figure 5ff

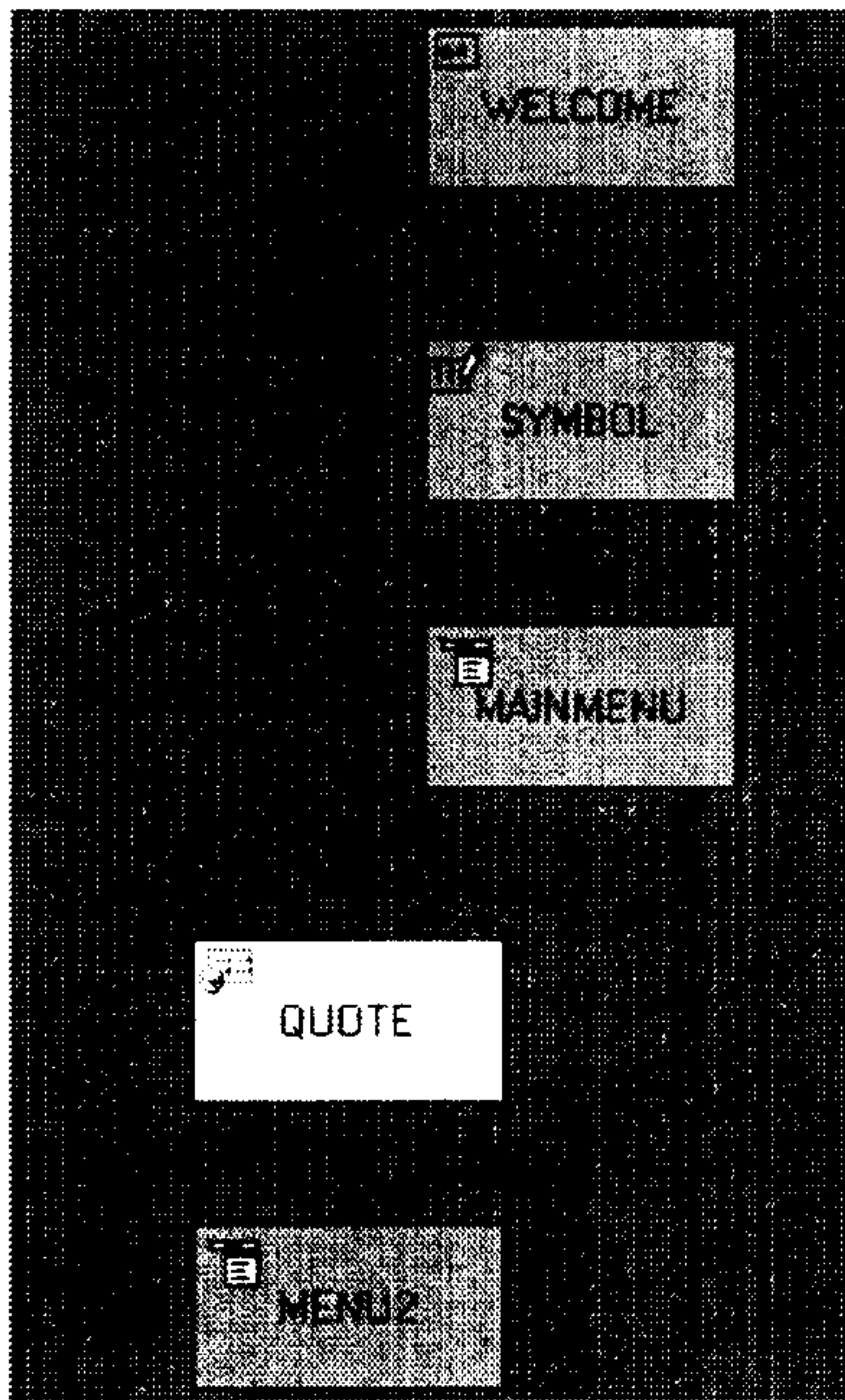


Figure 5gg

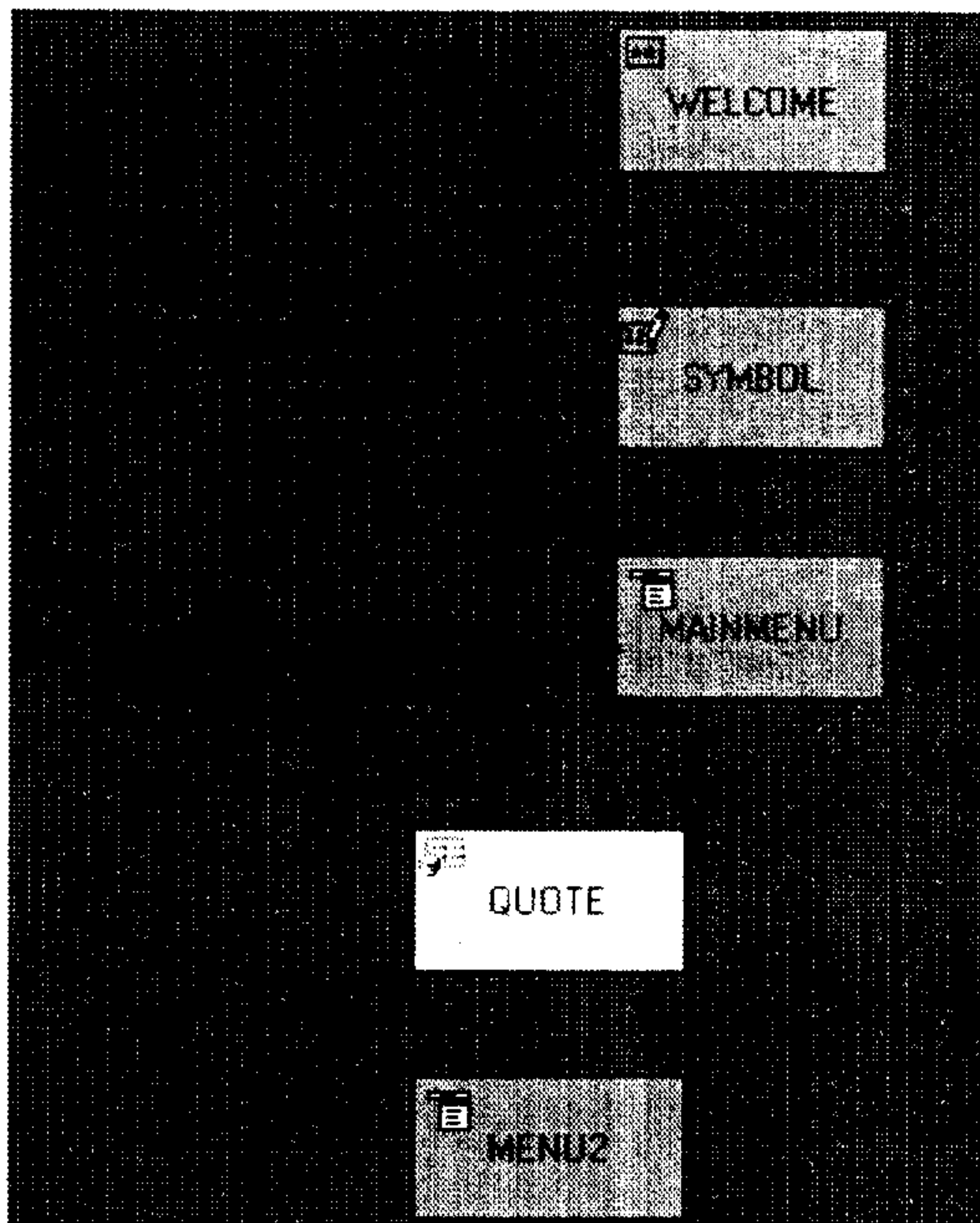


Figure 5hh

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages |

Data Source Login:

Data Source Password:

Data Source Name:

Table Name:

Return Action

Display Records Display as Options Display as Table

Return Record as Variable

Figure 5ii

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Data Source Login:

Data Source Password:

Data Source Name:

Table Name:

Return Action

Display Records Display as Options Display as Table

Return Record as Variable

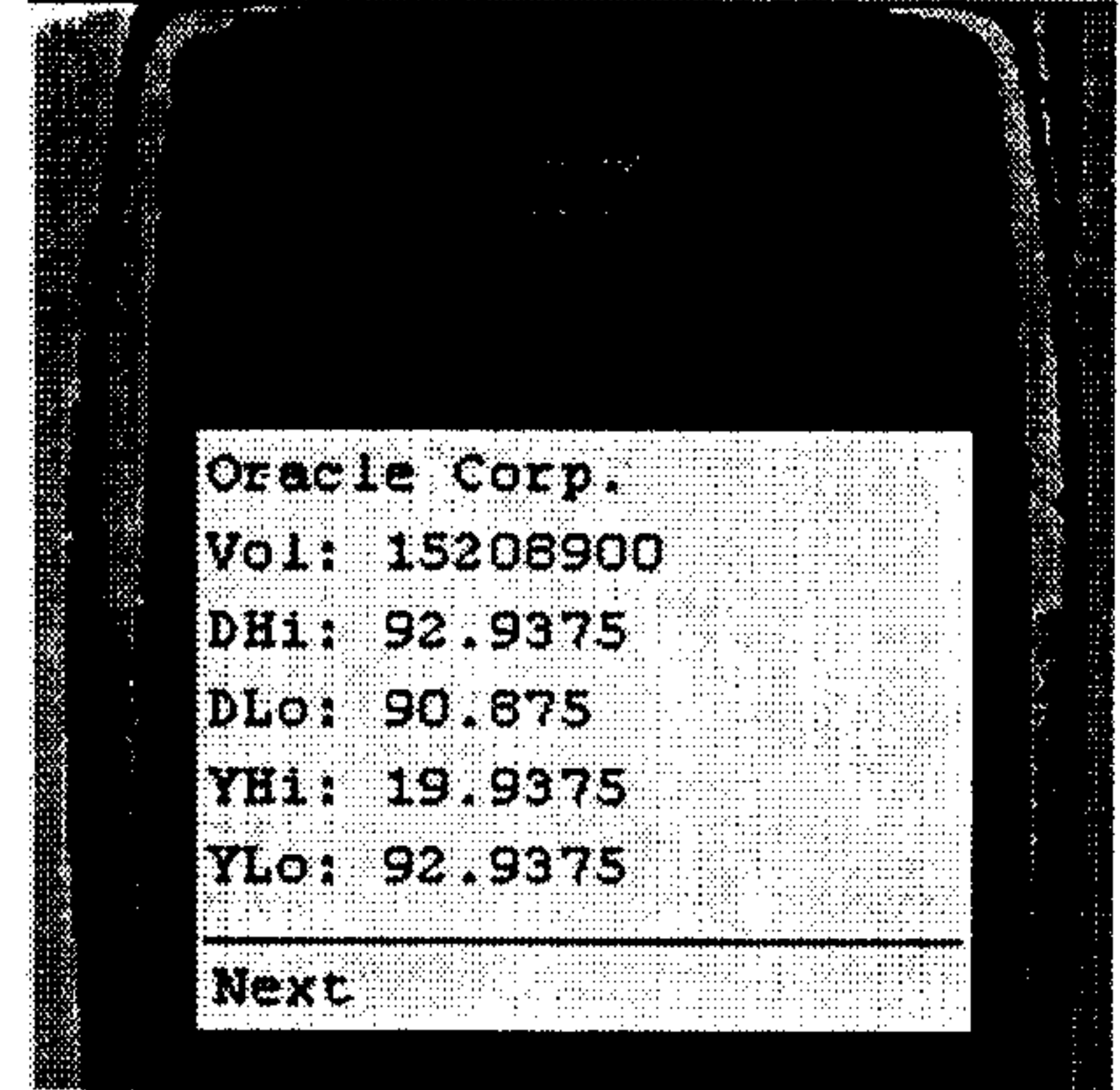


Figure 5jj

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Display Records Display as Options Display as Table

Field Name	Header	Type	Visible

Criteria:

Order by:

Return Record as Variable

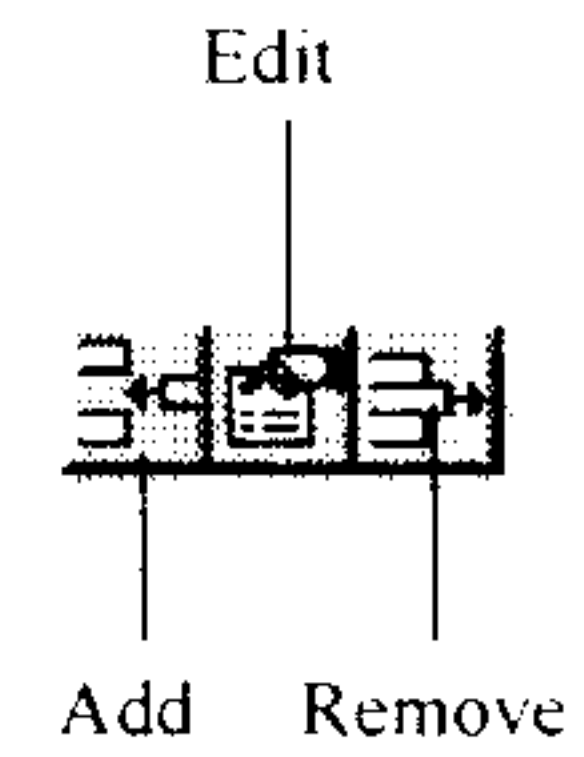


Figure 5kk

Add Field Information

Field Name:

Header:

Type: Alpha Numeric

Visible

Figure 5ll

Add Field Information

Field Name:

Header:

Type: Alpha Numeric

Visible

Figure 5mm

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Field Name	Header	Type	Visible
Name		Alpha	Yes
Volume	Vol. 	Numeric	Yes
DayHi	DHi. 	Numeric	Yes

Criteria:

Order by:

Return Record as Variable

Figure 5nn

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

ADO Connection String:

SQL String:

Return Record as Variable

Figure 5oo

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Operation Successful:

Operation Failed:

Error Encountered:
 Display Standard Message

Empty Record set:
 Display Standard Message

Return Record as Variable

Figure 5pp

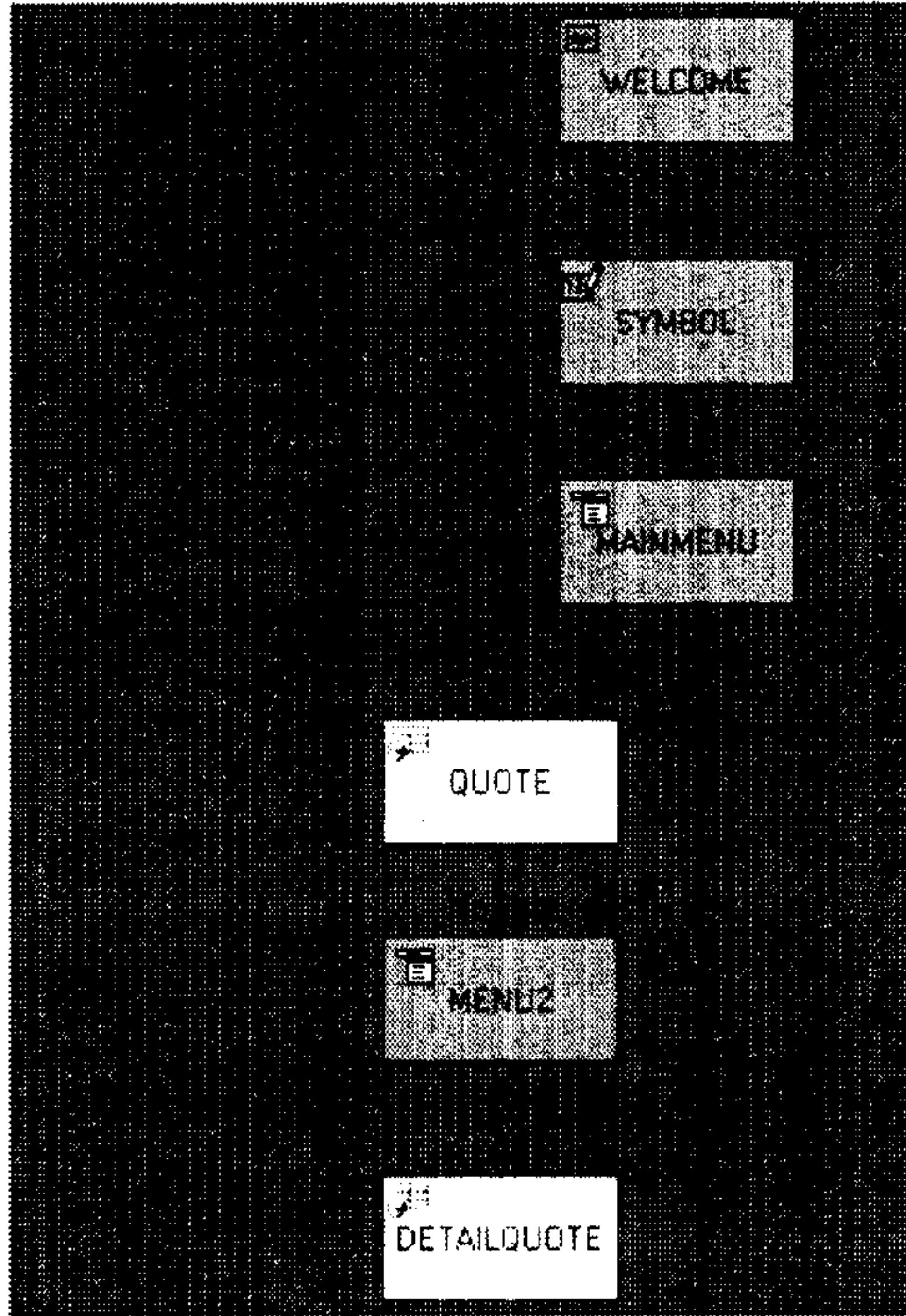


Figure 5qq

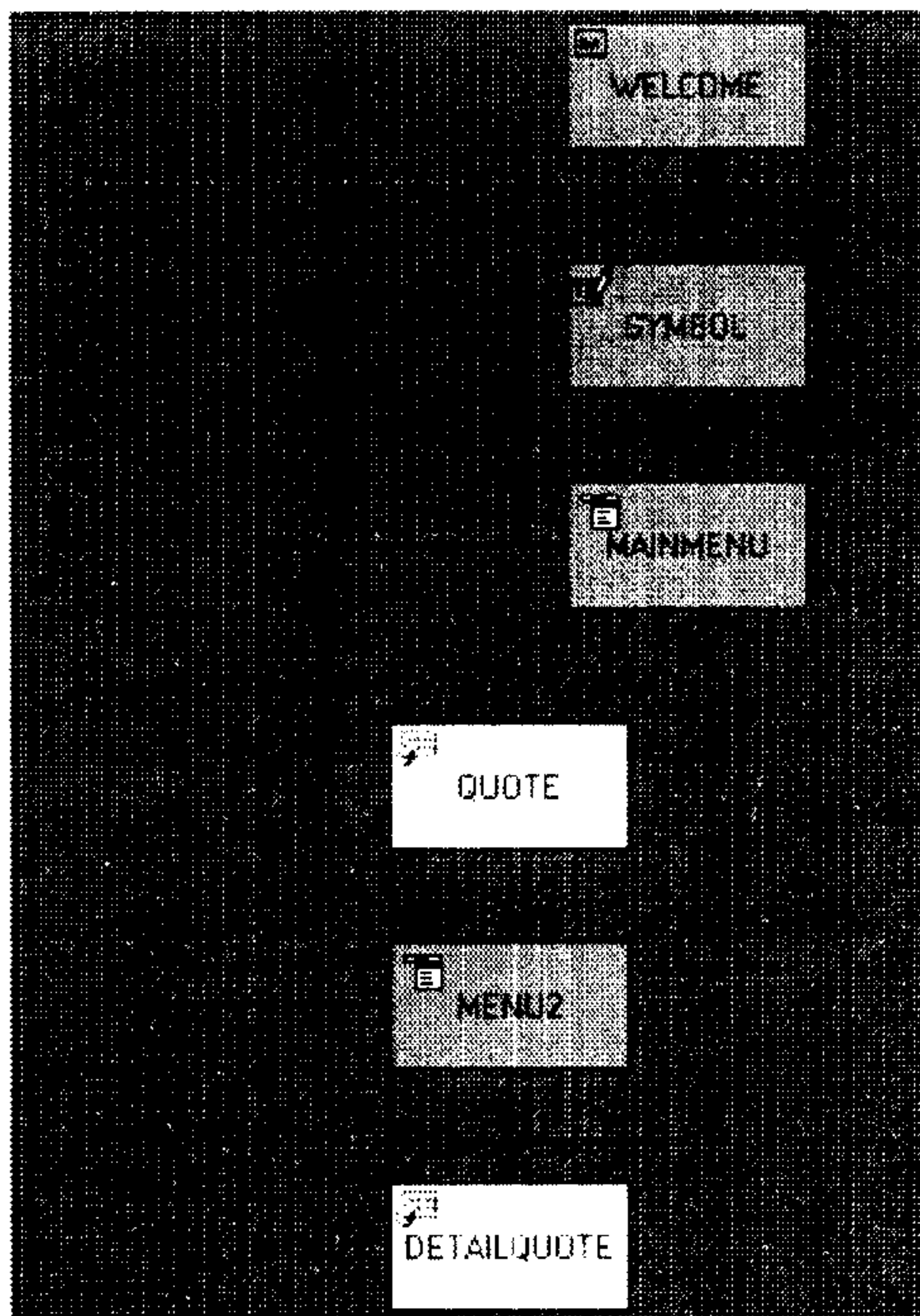


Figure 5rr

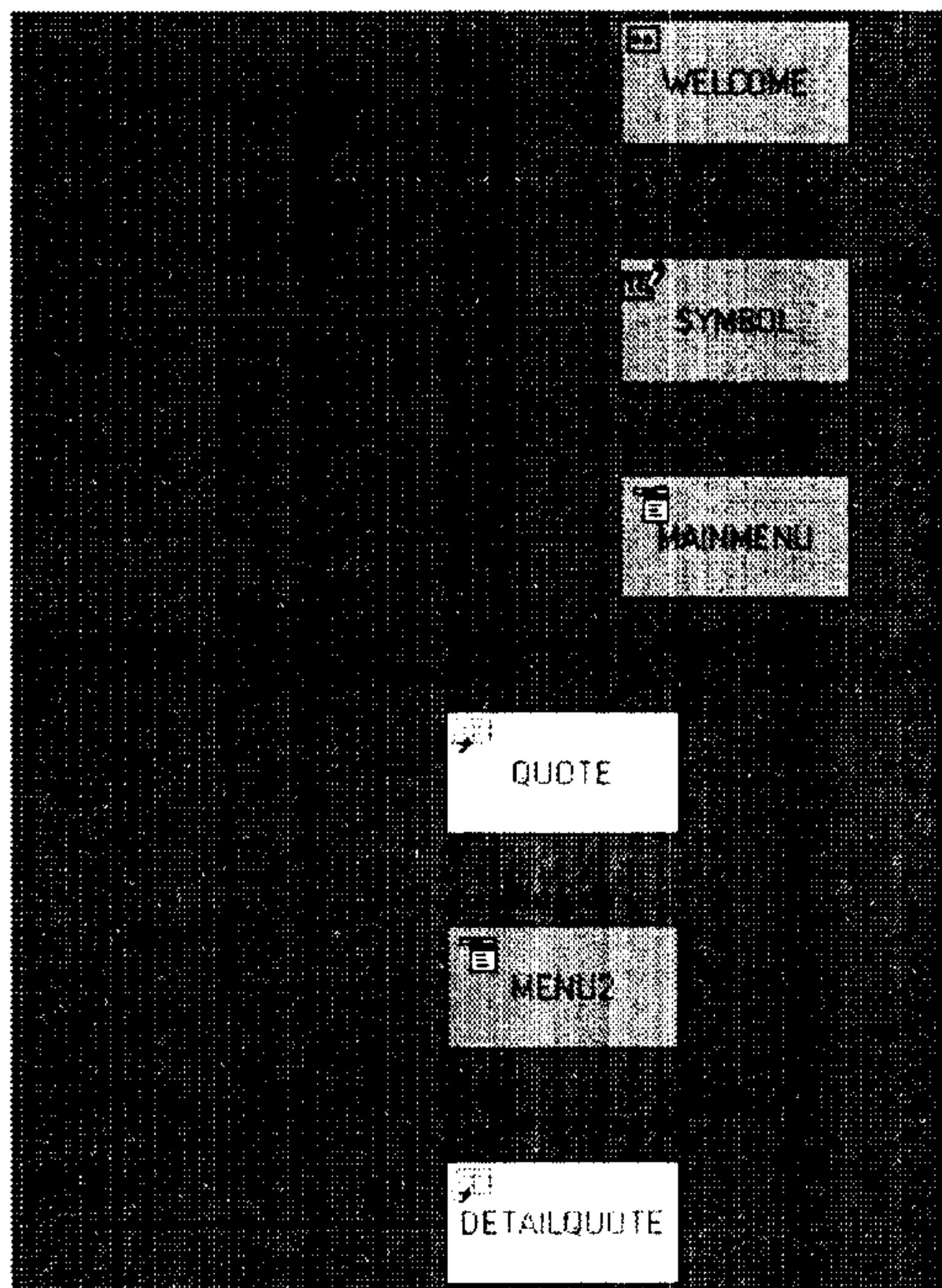


Figure 5ss

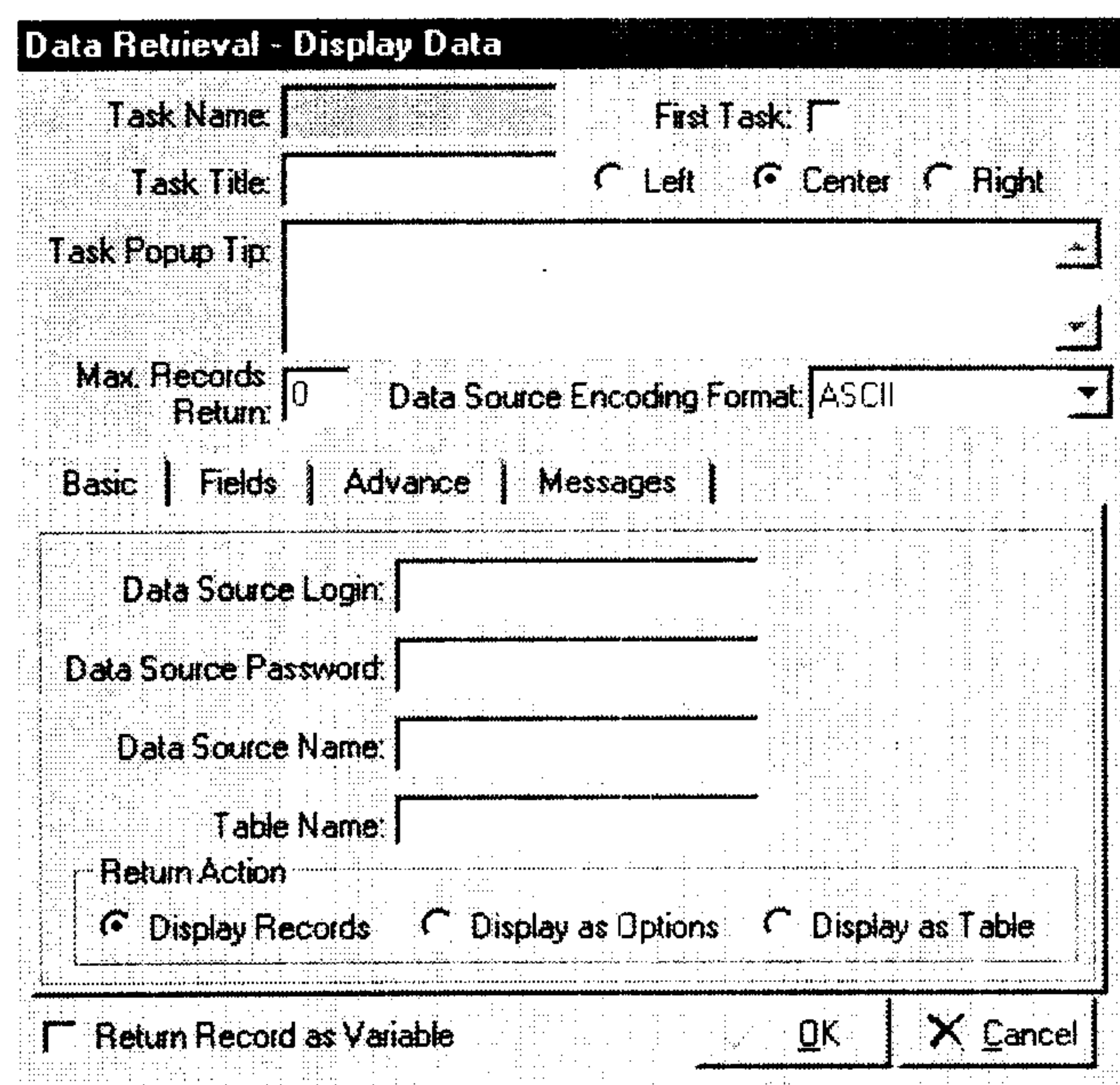


Figure 5tt

Data Retrieval - Display Data

Task Name: HEADLINE First Task:

Task Title: Left Center Right

Task Popup Tip: Display news headline

Max. Records Return: 0 Data Source Encoding Format: ASCII

Basic | Fields | Advance | Messages

Data Source Login: _____

Data Source Password: _____

Data Source Name: _____

Table Name: News

Return Action

Display Records Display as Options Display as Table

Return Record as Variable

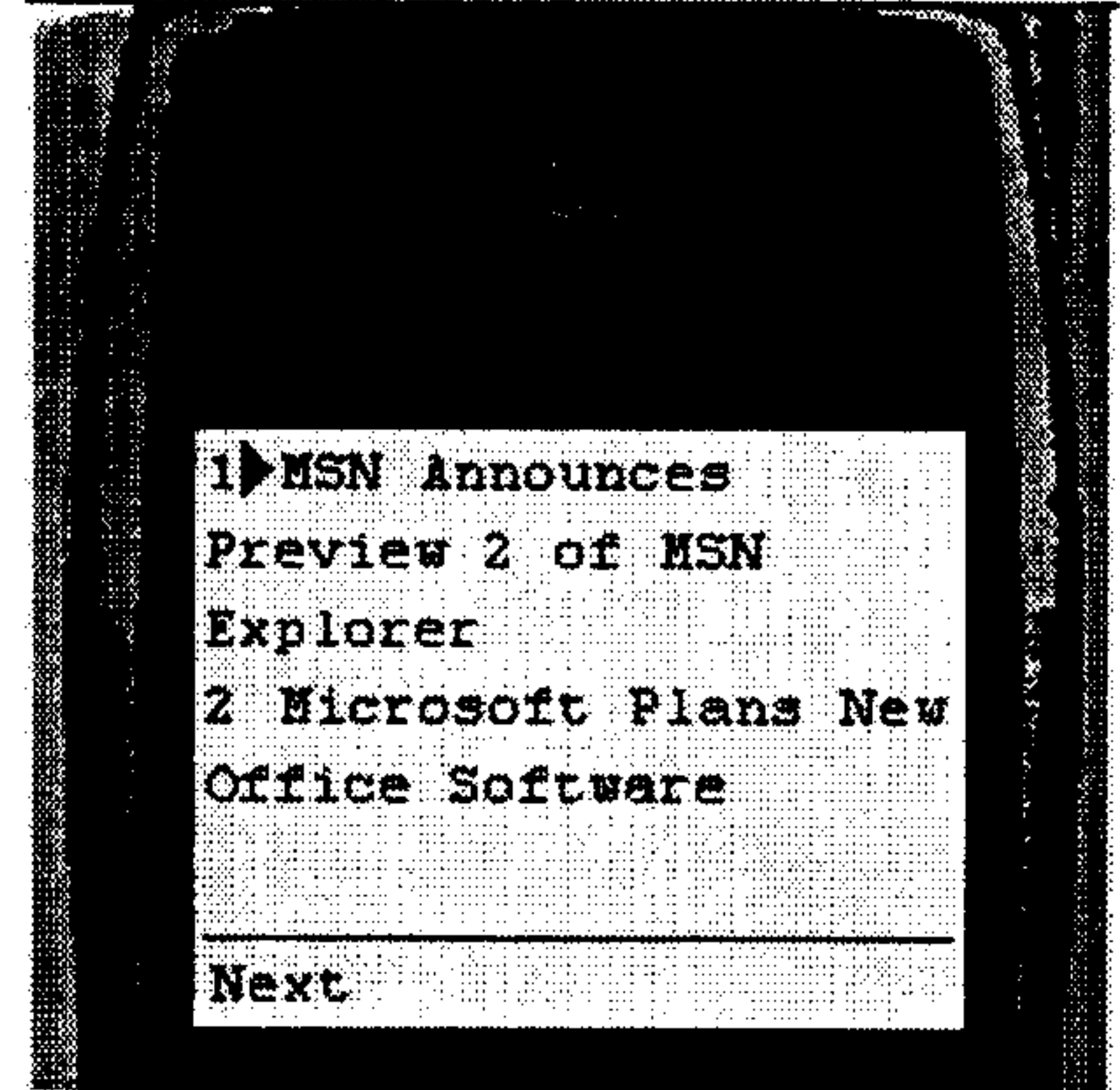


Figure 5uu

Data Retrieval - Display Data

Task Name: HEADLINE First Task:

Task Title: Left Center Right

Task Popup Tip: Display news headline

Max. Records Return: 0 Data Source Encoding Format: ASCII

Basic | **Fields** | Advance | Messages

Field Name	Header	Type	Visible

Criteria: _____

Order by: _____

Return Record as Variable



Figure 5vv

Add Field Information

Field Name:

Header:

Type: Alpha Numeric

Visible

Figure 5ww

Add Field Information

Field Name: RecordID

Header:

Type: Alpha Numeric

Visible

Figure 5xx

Add Field Information

Field Name: Headline

Header:

Type: Alpha Numeric

Visible

Figure 5yy

Data Retrieval - Display Data

Task Name: HEADLINE First Task:

Task Title: Left Center Right

Task Popup Tip: Display news headline

Max. Records Return: 0 Data Source Encoding Format: ASCII

Basic | Fields | Advance | Messages

Field Name	Header	Type	Visible
RecordID		Numeric	No
Headline		Alpha	Yes

Criteria: Symbol = '\$SYMBOL'

Order by:

Return Record as Variable OK Cancel

Figure 5zz

Data Retrieval - Display Data

Task Name: HEADLINE First Task:

Task Title: Left Center Right

Task Popup Tip: Display news headline

Max. Records Return: 0 Data Source Encoding Format: ASCII

Basic | Fields | Advance | Messages

ADO Connection String:

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=\$%YBOPATH\MDB\Stock.mdb

SQL String:

Return Record as Variable OK Cancel

Figure 5aaa

Data Retrieval - Display Data

Task Name: HEADLINE First Task:

Task Title: Left Center Right

Task Popup Tip: Display news headline

Max Records Return: 0 Data Source Encoding Format: ASCII

Basic | Fields | Advance | Messages

Operation Successful:

Operation Failed:
Failed to get news headline

Error Encountered:
Error getting news headline Display Standard Message

Empty Record set:
No news Display Standard Message

Return Record as Variable OK Cancel

Figure 5bbb

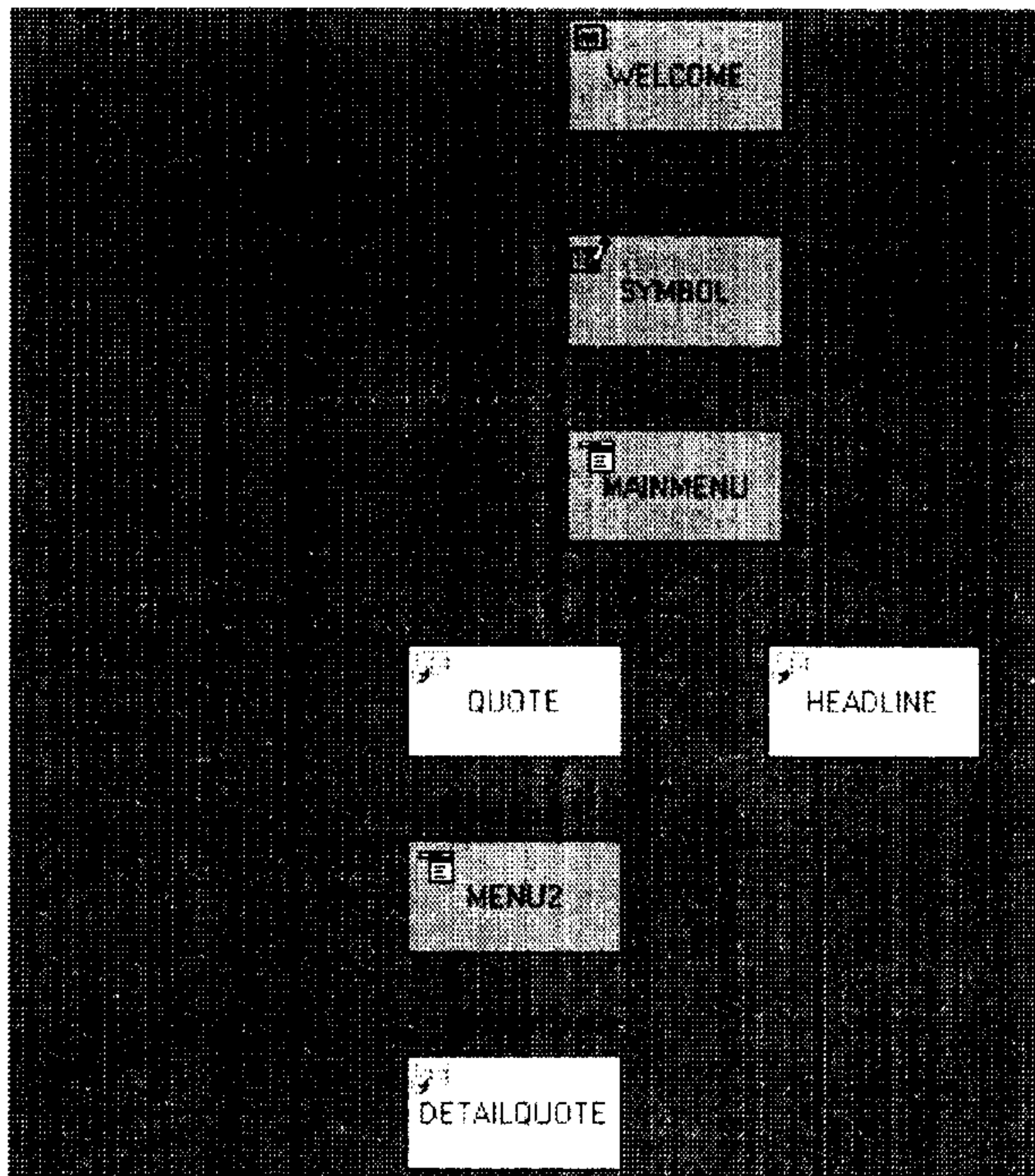


Figure 5ccc

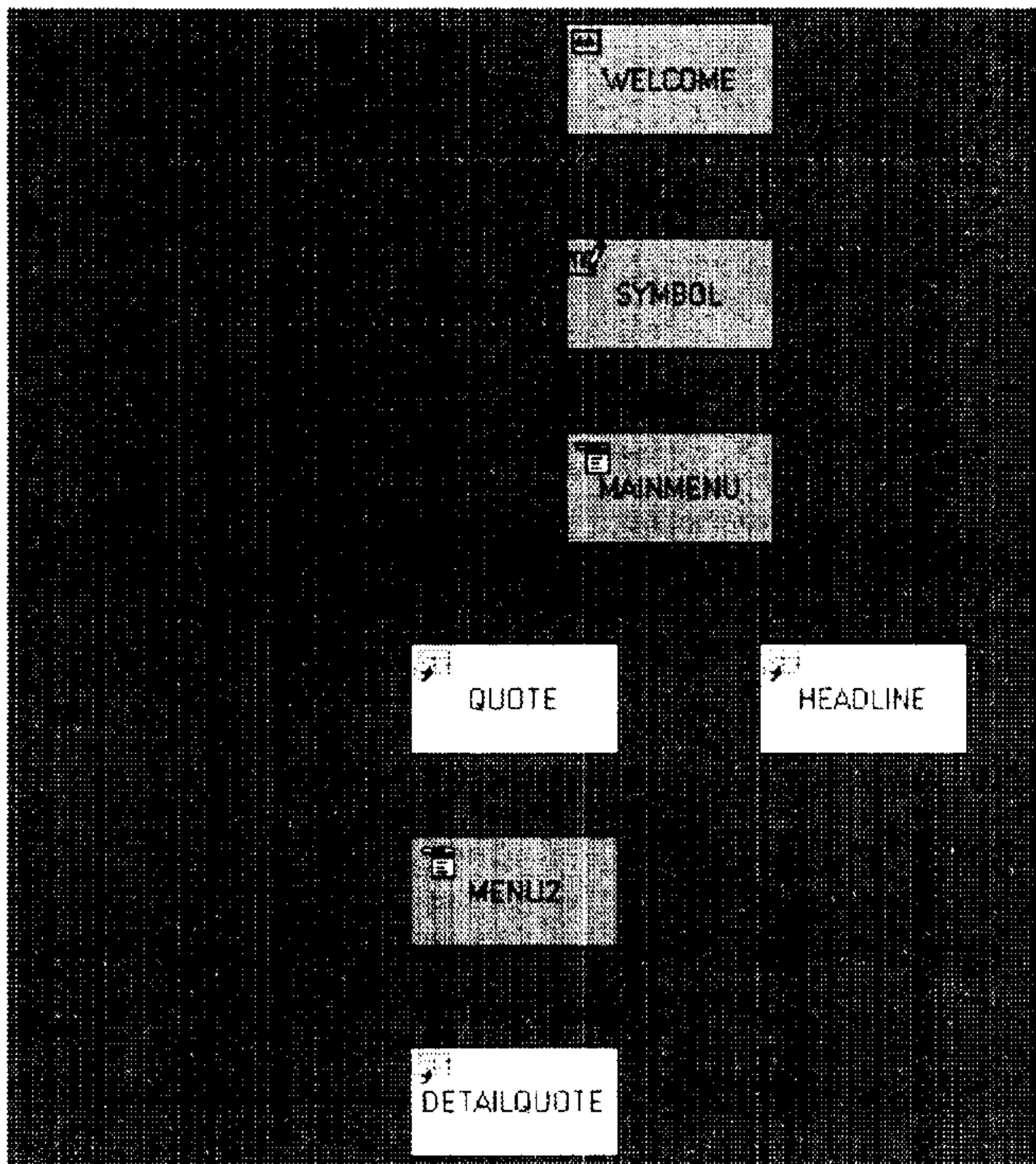


Figure 5ddd

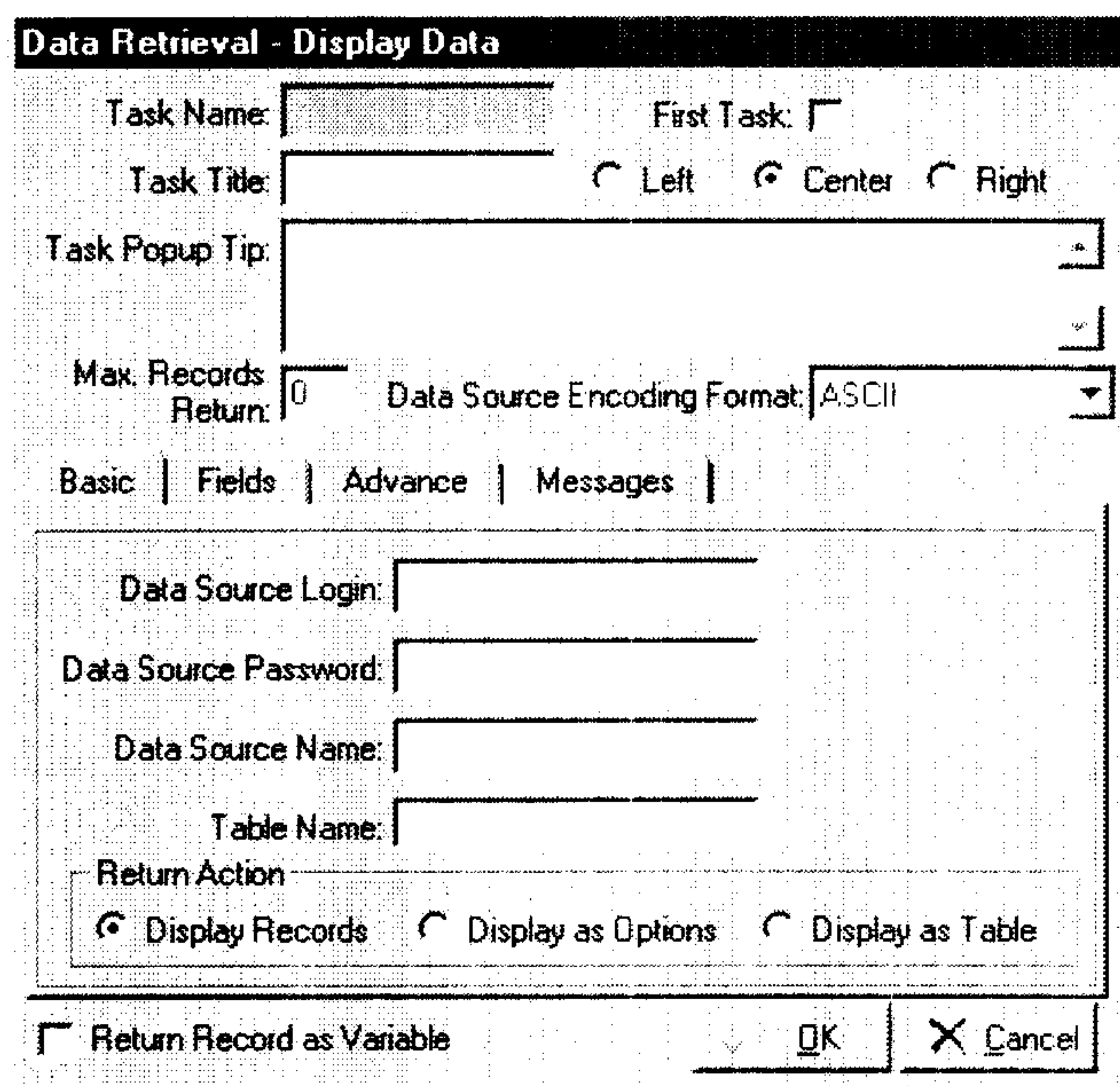


Figure 5eee

Data Retrieval - Display Data

Task Name: First Task

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Data Source Login:

Data Source Password:

Data Source Name:

Table Name:

Return Action

Display Records Display as Options Display as Table

Return Record as Variable

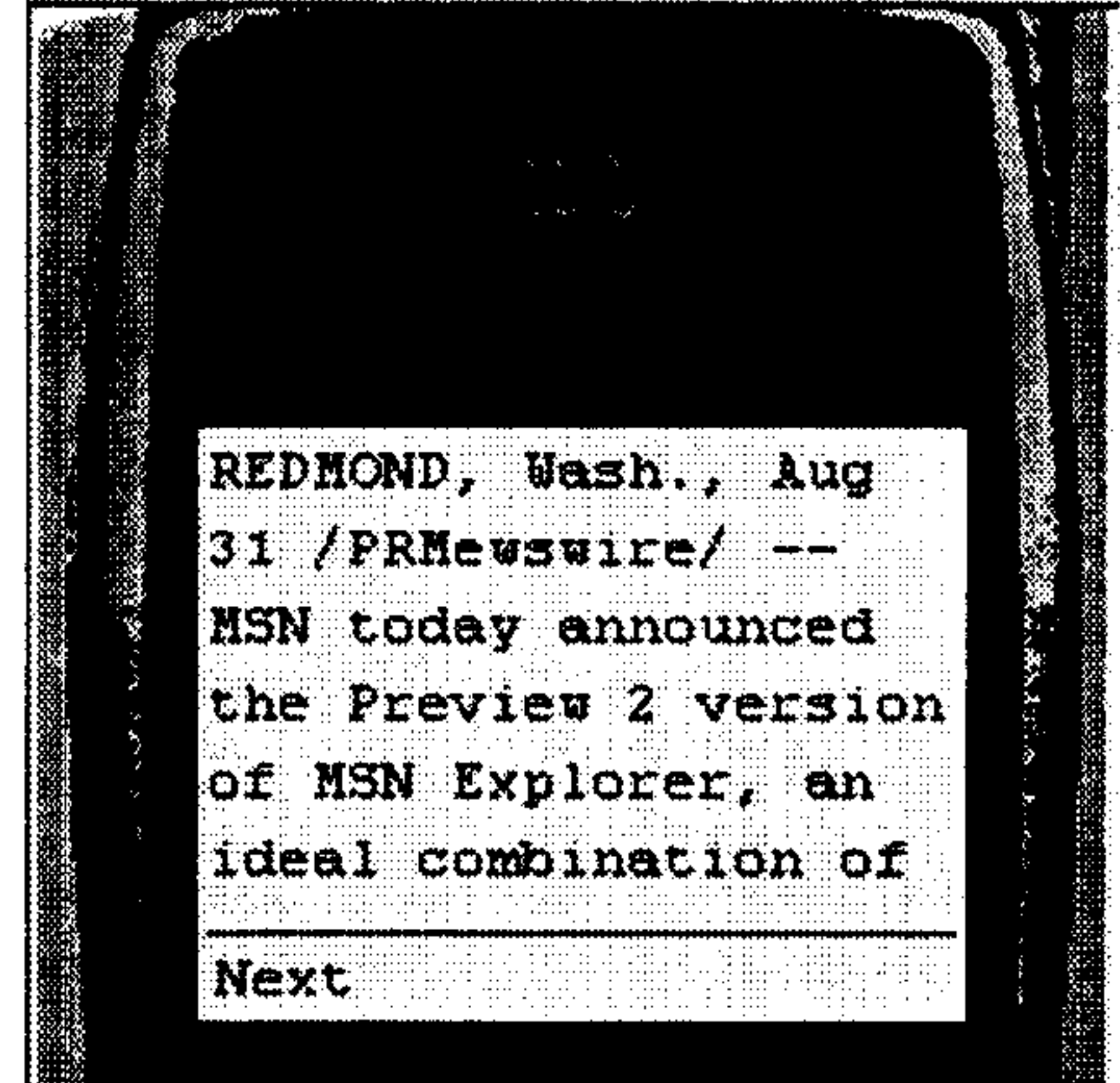


Figure 5fff

Data Retrieval - Display Data

Task Name: First Task

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Field Name	Header	Type	Visible

Criteria:

Order by:

Return Record as Variable



Figure 5ggg

Add Field Information

Field Name:

Header:

Type: Alpha Numeric

Visible

OK Cancel

Figure 5hhh

Add Field Information

Field Name:

Header:

Type: Alpha Numeric

Visible

OK Cancel

Figure 5iii

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages

Field Name	Header	Type	Visible
Detail		Alpha	Yes

Criteria:

Order by:

Return Record as Variable OK Cancel

Figure 5jjj

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages |

ADO Connection String:

SQL String:

Return Record as Variable

Figure 5kkk

Data Retrieval - Display Data

Task Name: First Task:

Task Title: Left Center Right

Task Popup Tip:

Max. Records Return: Data Source Encoding Format:

Basic | Fields | Advance | Messages |

Operation Successful:

Operation Failed:

Error Encountered:
 Display Standard Message

Empty Record set:
 Display Standard Message

Return Record as Variable

Figure 5lll

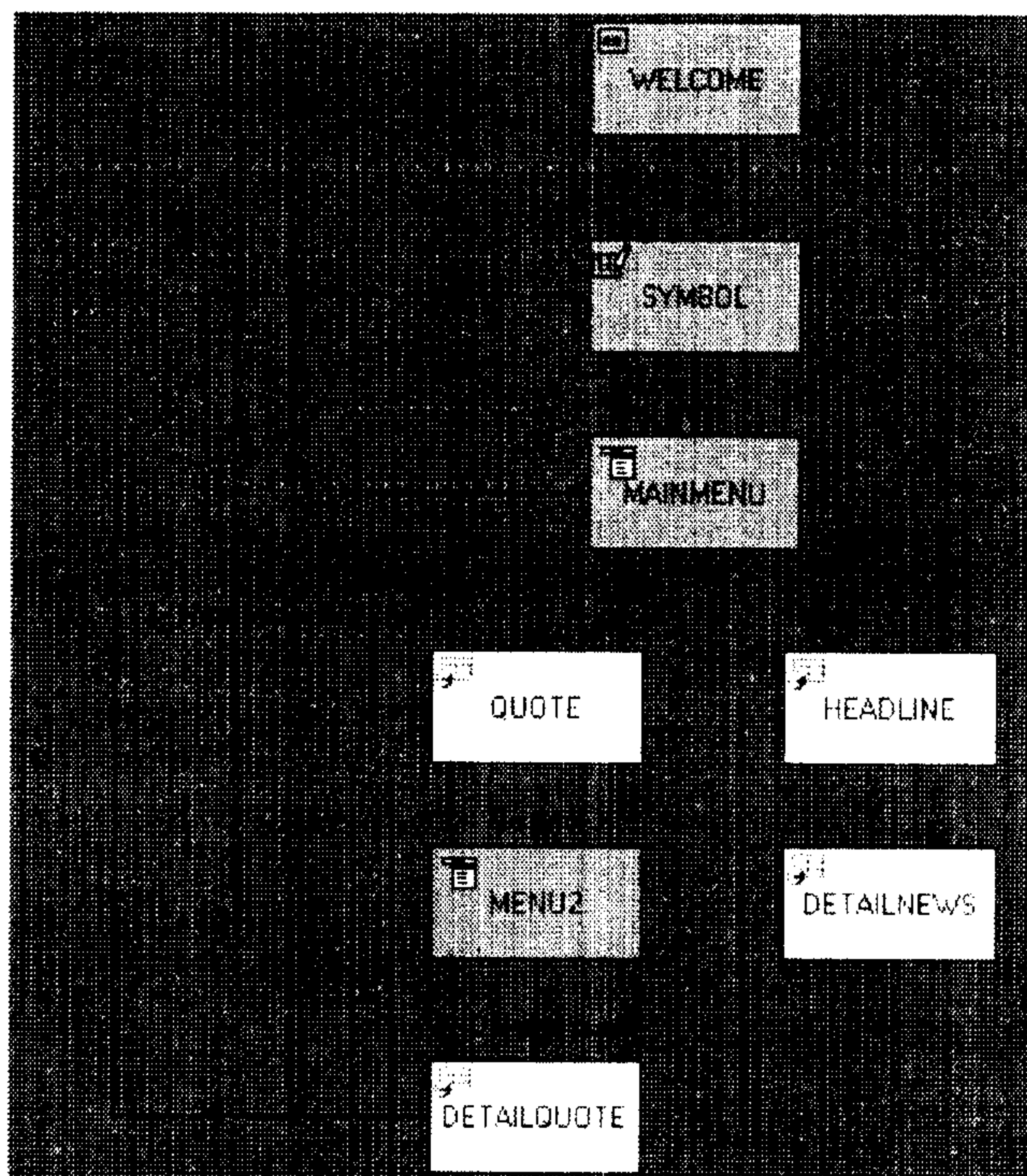


Figure 5mm

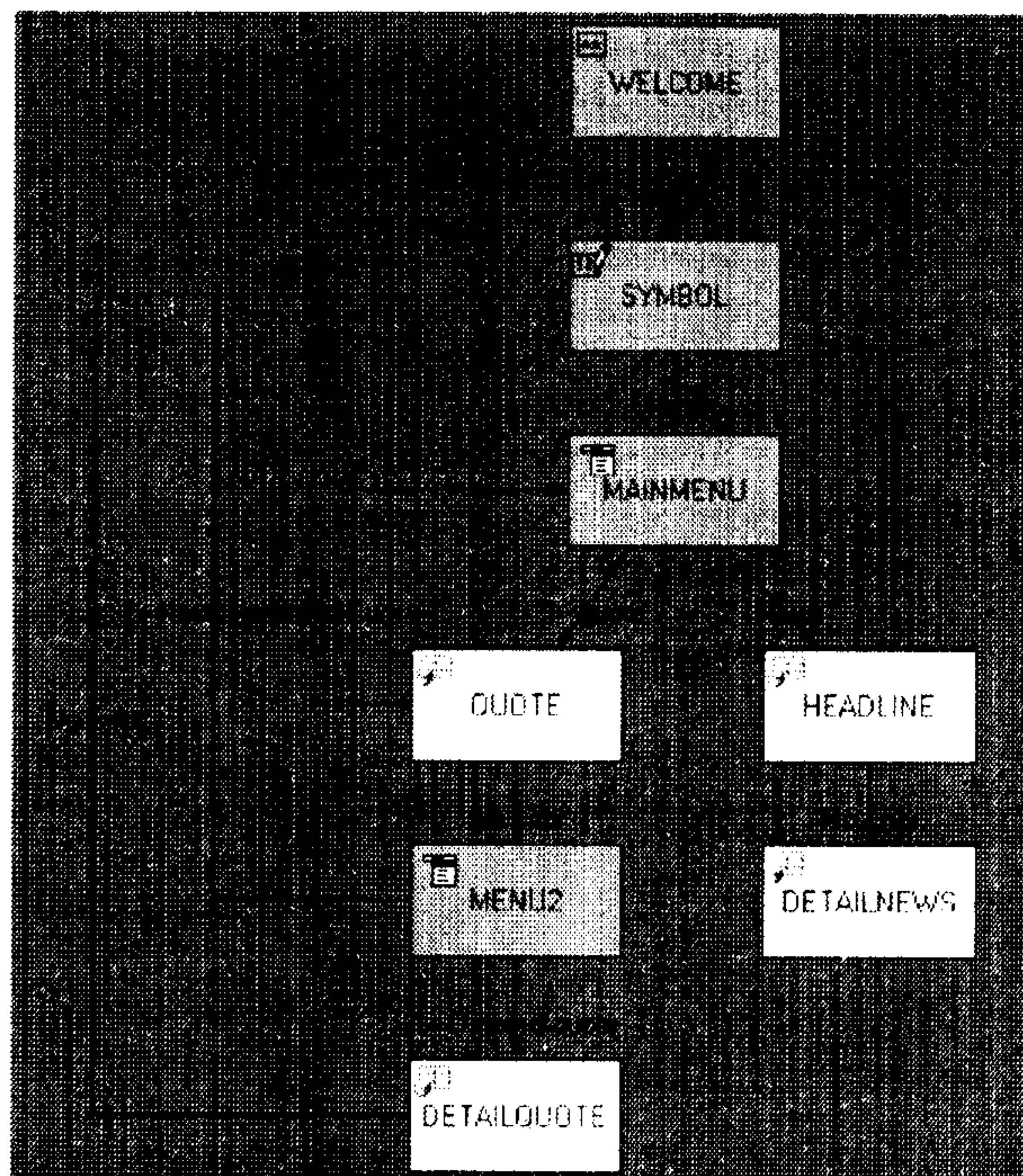


Figure 5nn

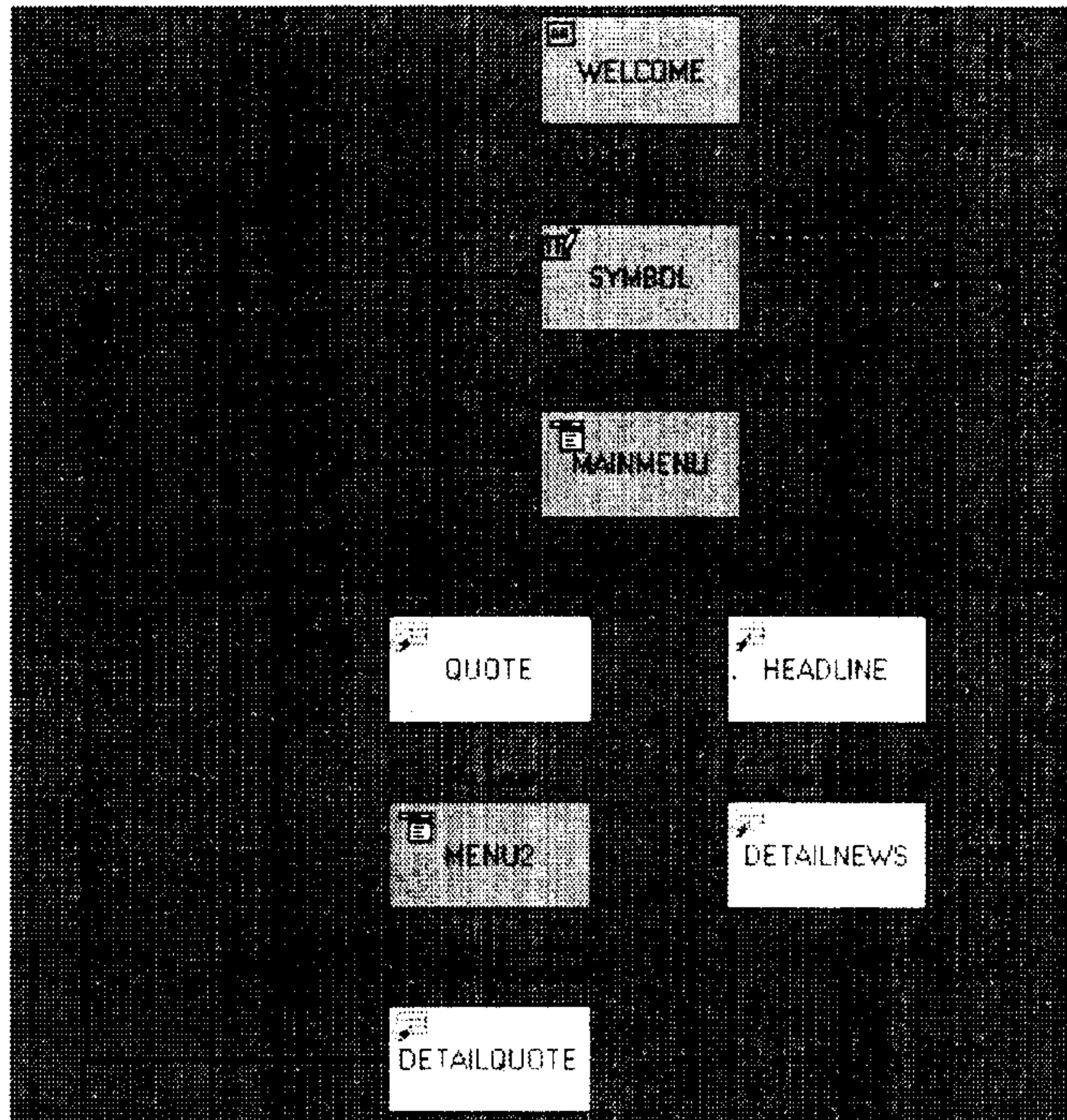


Figure 5000

