PhoneSoft

Software Developer’s Kit

Second edition 2003.

Domino Telecom Solutions.
www.phonesoft.com

Contents

Introduction

The PhoneSoft SDK (Software Developer's Kit) includes tutorials, Help files and sample applications.

This *PhoneSoft SDK User Guide* includes instructions for using the PhoneSoft Script Generator and descriptions of the available script commands.

The book also has instructions for using the License File Generator and the HelpDesk and FaxBack sample applications.

CHAPTER 1: Script Generator

Using the PhoneSoft Script Generator

You can use the PhoneSoft Script Generator to create Notes telephony applications. The script generator creates or edits a script file. A script file is a sequence of commands that defines the flow of a phone call.

This section describes how to start the Script Generator, to open a script or create a new script.

For a list of the script commands that are included with this package, see "Script commands" on page 19.

To start the Script Generator

1. Start Phone Server.
2. Click the Edit>MailCall>Script menus.

To open an existing script

1. Start the Script Generator.
2. In the Select MailCall Script dialog box, click "Browse" and locate the script file name.
3. Select the file name and then click "Edit" to open it.

To create a new script

1. Start the Script Generator.
2. In the Select MailCall Script dialog box, confirm that the Script File box is blank and click "Edit."
3. Click the File>Save menus and then name and save the file. The file can be saved in any directory.
4. Confirm the file path is correct in the Script File list.
5. In the Application section of Phone Server, confirm that "UMailCall" is the selected application for the phone lines for which you are creating the script file.
6. Double-click the "UmailCall" application and in the Line Setup dialog box, confirm that the "Auto Start" check box is selected and click "Close." With "Auto Start" set, Phone Server automatically runs the selected application on that phone line at startup.
7. Begin creating your script for your application as appropriate.

Using script commands

After creating a script, you can add commands or change existing commands. You can also rename or delete commands from a script.

When creating a script, the New Child menu assigns a command as subordinate to the selected command in the script's tree structure.

To add a script command or child

1. Open the Script Generator.
2. Open the script for which you want to add a command.
3. Right-click the existing command to appear before the new command. If there are no commands, click the root directory.
4. To create a new command, click "New" from the pop up menu.

   Or to create a new child, click "New Child."
5. In the Select New Item dialog box, select the appropriate command type from the list and click "OK."
6. In the script directory tree, confirm that the new command appears after the selected command, or the new child appears as the last child.

To edit an existing script command

1. Right-click the command to edit.
2. Select the Properties menu.
3. Select the command to edit and then press ENTER.
4. In the Properties dialog box, edit the command as needed.

To rename an existing script command

1. In the script directory tree, right-click the command to rename.
2. Select the Rename menu and type the new name.

   NOTE: *Press ESC to cancel the rename.*

To delete an existing script command

1. Right-click the command to delete.
2. Click the Delete menu. The command and any associated children are deleted.

CAUTION: *Deletions are final; no delete confirmation dialog box appears.*

Using script variables

When a script command field specifies a variable, the variable may be one of three types: a quoted string, a Notes text field value, or a reserved word. Script variables are case sensitive.

Reasons for using variables

- Names and paths are simple. For example, “sDocDB” as compared to “C:\Phonesoft\HelpDeskDemo\hdeskdat.nsf.”
- Updates and changes are easy because you can make a single change instead of searching for every instance in the application to change.

Quoted strings

Quoted strings are interpreted as literal values with the quotes ignored. No escape code can be used with quotes in the string. A variable inside {} (curly brackets) is replaced with the variable value. If the variable inside the brackets is a numeric variable, it is converted to text with the number of decimal places specified by the Significant_digits reserved word.

For example, the “sForm” variable is replaced with “memo” in the following command lines:

```
sForm = ‘memo’

sSelectionFormula = ‘Select Form = ‘{sForm}’’
```

Notes text fields

When a LotusScript command runs, it may modify Notes text fields in an in-memory Notes document. You can use these Notes text fields as variables in your script commands. LotusScript commands are not case sensitive.

Reserved words

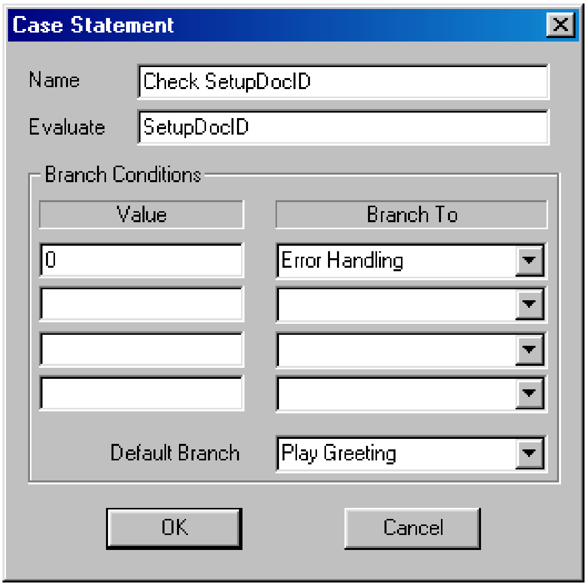
The following reserved words are supported:

Reserved word	**Description**
AGENT_DATABASE	The database where Lotus agents are run by using LotusScript.
AGENT_SERVER	The server where Lotus agents are run by using LotusScript.
ANI_INFORMATION	If available, the ANI (automatic number identification) information for a call.
APPLICATION	This is the name of the application that is running. It is the string taken from the Application column in Phone Server.
CALL_TYPE	If available, a numeric value representing the type of a call: ▪ 0 is an outside or direct call ▪ 1 is a forwarded, unanswered call ▪ 2 is a forwarded, busy call ▪ 3 is a forward of all calls
CALLER_ID	The phone number string from the raw Caller ID information.
DATE_NOW	A read-only value for assigning the current date and time to a Notes field.
DIGITS_STRING	String of digits retrieved with a “Get digits” command. When a caller presses a termination key, it is stored in the “Term_digit” reserved word. This string will not contain a termination key.
EMPTY	A read-only value for clearing a Notes field.
ERROR_BRANCH	Runs when an error occurs. The default is “MailCall goodbye.”
INBAND_STRING	String of digits retrieved with a “Get inband” command.
LINE_NUMBER	Numeric value of the phone line number that is running the application.
MENU_KEY	The most recent menu key that a caller pressed from the DTMF Menu.
PHONE_STRING	Phone number assigned to the line running the application. This string is not the PBX phone number, but the number used by the administrator to assign the different phone lines and to sort applications in a script.
RAW_CALLER_ID	The Caller ID string as it appears to the Dialogic board.
SIGNIFICANT_DIGITS	The number of significant digits used when converting numeric entries to text. The default number of significant digits is 0.
SMDI_CALL_SOURCE	If provided by the phone system, the source of the call.
SMDI_CALLED_NUMBE R	If provided by the phone system, information on the called party.
SMDI_CALLING_NUMB ER	If provided by the phone system, information on the calling party.
TERM_DIGIT	Termination character, either ✶ or #, from a Get Digits command or a Record File command.
VOICE_FILE	The current voice recording from a “Record file” command.

Using branch variables

A branch variable is either the “Name” field of another script command, a reserved word, a MailCall entry point, or a string variable. The “Name” field is case sensitive. If the branch variable is a MailCall entry point, there will be no associated “MailBox” field.

Usually used in the “ThenGoTo” field or the “Default Branch” field in a script command. For example, in the following screen the branch variable is Play Greeting. For details, see “Play Greeting” on page 56.



Case Statement

Name: Check SetupDocID

Evaluate: SetupDocID

Branch Conditions

Value	Branch To
0	Error Handling

Default Branch: Play Greeting

Beginning with an outline

The most difficult part of creating an application is to begin the script design. The best way to start is to create a brief outline for your application. The outline can include what the callers hear first and then what menu options are offered and where they lead the caller to next, and so on.

For example, the following is an outline of the HelpDesk application. For script details, see “HelpDesk script” on page 49.

After creating the outline, you are ready to begin creating the script with the Script Generator. Start simple with the basic commands and write the command names next to the applicable step in your outline.

HelpDesk script outline

Application setup

Plays greeting Plays an opening greeting, such as “Thank you for calling the XYZ HelpDesk system. Press 1 to leave a help request. Press 2 to check on the status of an existing request.”

Plays main menu Pressing 1 sends callers to “Leave Help Request” and pressing 2 sends callers to Check Request Status.

1 for Leave Help Request Collects the following information for the request:

1. Requests extension
2. Records request
3. Requests priority
4. Records the status number
5. Stores message in database
6. Plays status number back to caller and then increments the number for the next caller
7. Returns call to main menu

Continued

2 for Check Request Status Asks for the following information when callers are checking requests.

1. Requests status number
2. Checks for response:
   - If a caller chooses “Yes,” plays responses.
   - If “No response,” plays a no response message.
3. Returns call to main menu.

3 for Admin Only available for administrators to check and reply to requests. Collects the following information:

1. Requests Admin password
   - Validate password
2. Requests status number
3. Plays request
4. Records response to request
   - 1 for yes, record response and store message in the database
   - Any other key, return to 2 for Enter Status Number.

Goodbye Plays a goodbye message and ends the call.

Using the Phone Server log

After creating an application, you can use the log file to test and debug your script file. In the lower half of the Phone Server window pane is the activity log. You can set this log to show only errors, activity or diagnostics (which includes everything). You can view different information in the log by using the “Error Only,” “Activity” or “Diagnostics” toolbar buttons.

Click the “View log” toolbar button to print the log activity into a text document. The log text document contains:

- Date and time of each action,
- Line number,
- Application running,
- Log level, and
- Brief description of the activity.

Any errors display as bold, red text and activities display as blue text. System activities display as black text.

Using the License File Generator

You can use the License File Generator to create a license file (an encrypted *.lic text file) for distributing and protecting your custom applications.

The script file is created with the SDK. The application license file is created with the License File Generator. The license file contains the following:

- License number type.
- Number of enabled ports for the application.
- Encrypted number, which prevents a customer from editing the license file content.
- Application name that appears in Phone Server.

The generator is provided and can be used with the PhoneSoft SDK. The SDK includes a two-port license file intended for internal development only.

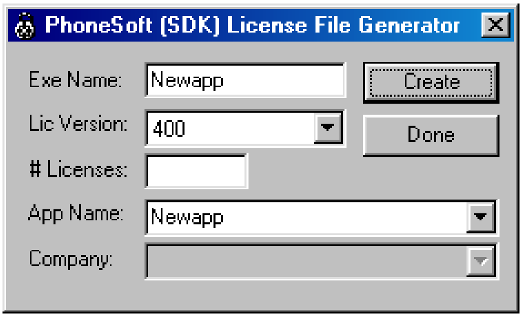
A runtime license is required for distributing a customized application. For example, to distribute an application named Newapp, use the License File Generator to create a Newapp.lic file.

The License File Generator creates a 400 Type license file, such as Newapp.lic. However, for an application to run correctly, the Phone Server computer must have both the 400 Type license file and a 300 Type PhoneServer.lic file, which is included with the PhoneSoft software.

With the SDK, application files are forms (PSS) files, which are sequences of forms run by Phone Server. The SDK includes the HelpDesk.pss and Faxback.pss sample application files and a Type 300 Phoneserver.lic file, which restricts each application to two ports.

To create a license file

1. Locate and double-click the Pslicense.exe file.
2. In the PhoneSoft License File Generator dialog box, complete the fields, click “Create,” and then follow the on-screen instructions.



PhoneSoft (SDK) License File Generator

Exe Name: Newapp

Lic Version: 400

# Licenses:

App Name: Newapp

Company:

Exe name The application’s file name, which names the license file the same name. The application and license file names must match. You can use up to 16 characters in this field. For example, for HelpDesk.pss, type `HelpDesk`.

Lic version Set as “400,” which is the type of file generated.

# Licenses Set equal to the number of ports that a customer has purchased.

App Name Application name that is stored and displayed in the Phone Server Application list. You can use up to 16 characters in this field.

Company For display only. The company name is defined by the PhoneServer.lic file.

CHAPTER 2: Script commands

Assign Variables

Assigns values to local variables. The two types of variables supported are text and numeric. Each assign statement has the following format:

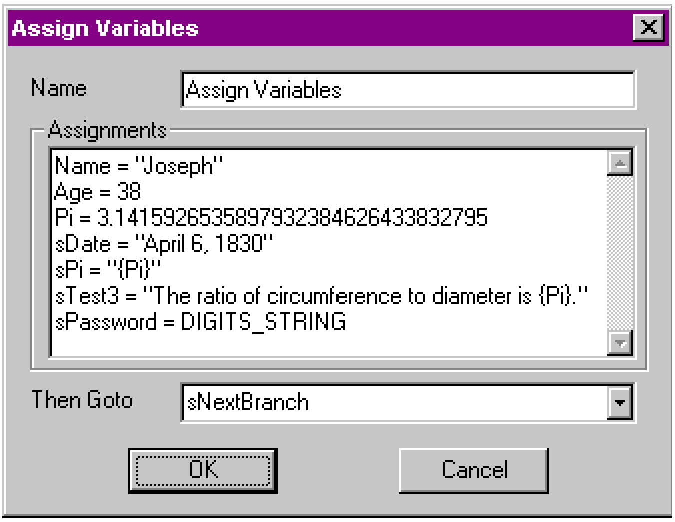
```
<variable> = <variable_value>
```

Where `<variable>` is the name of the variable that contains the value and `<variable_value>` is the value assigned to the variable. The variable value is a script variable.

If the variable already exists, the variable value is converted to the existing variable type.

If the variable does not exist, it is created with the assigned variable value.

NOTE: *Anything on the same line after a single quote will be ignored, which is useful for making comments.*



Name: Assign Variables

Assignments

```
Name = "Joseph"
Age = 38
Pi = 3.14159265358979323846264338327950
sDate = "April 6, 1830"
sPi = "{Pi}"
sTest3 = "The ratio of circumference to diameter is {Pi}."
sPassword = DIGITS_STRING
```

Then Goto: sNextBranch

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Assignments The assignment commands that are run by the script. The variables are on the left of the = (equal) sign and the variable values are on the right.

Then Goto The branch variable that is used after the command runs.

Assign Variables examples

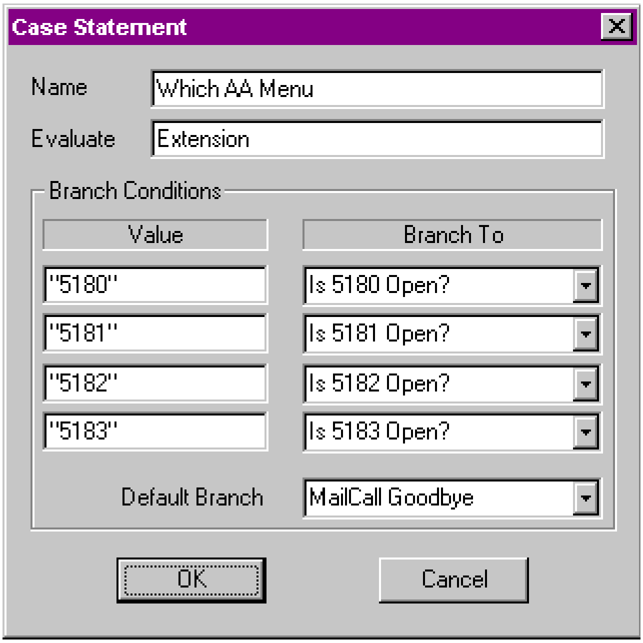
Assign variables:	Represents:
Name = “Joseph”	Text variable named Name with a value of “Joseph.”
Pi = 3.1415926535897932384626433832795	Numeric variable named Pi.
sPi = “{Pi}”	Text variable named sPi with a value of 3.14.
sPassword = DIGITS_STRING	Text variable named sPassword with the value of DIGITS_STRING, which is a reserved word.

***NOTE:** The number of significant digits converted in the sPi variable depends on the SIGNIFICANT_DIGITS reserved word. For details, see “Reserved words” on page 11.*

Case Statement

Evaluates a variable and branches to another command based on the variable's value. You can specify up to four branches and a default branch in a Case Statement.

For less than four branches, leave the unused fields blank. For more than four branches, you can cascade multiple Case Statements together.



Case Statement

Name: Which AA Menu

Evaluate: Extension

Branch Conditions

Value	Branch To
"5180"	Is 5180 Open?
"5181"	Is 5181 Open?
"5182"	Is 5182 Open?
"5183"	Is 5183 Open?

Default Branch: MailCall Goodbye

Name Unique name for this command. This name is used by other commands to determine where to branch to next.This

Evaluate A script variable and the value to be evaluated.

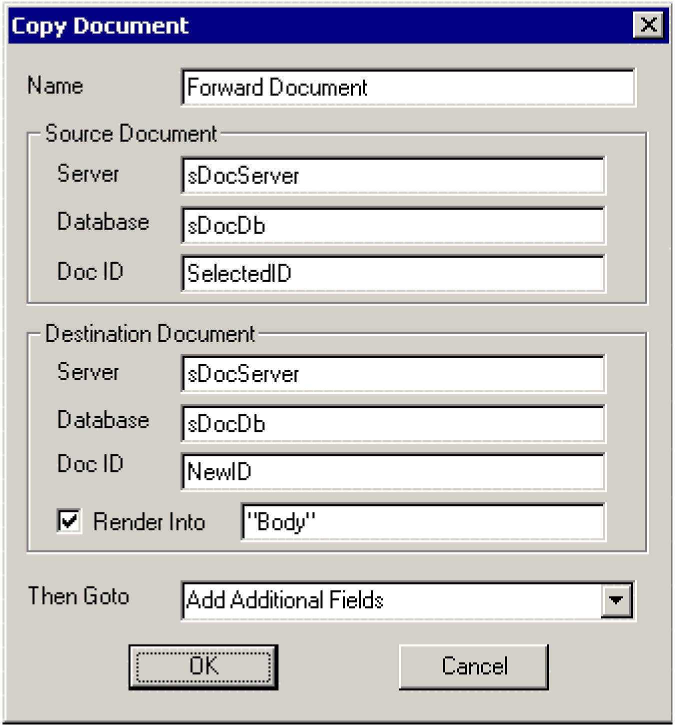
Value and Branch To Four values and their corresponding branches. Each "Value" field is a script variable and each "Branch To" field is a branch variable.

Default Branch The default is used when the variable is not equal to any of the other four branch values.

Copy Document

Creates a copy of a document in the specified database. You can copy the document in two ways:

- Create an exact copy of the document with all the same fields.
- Copy the document in rich text as the only field in the created document.



Name: Forward Document

Source Document
Server: sDocServer
Database: sDocDb
Doc ID: SelectedID

Destination Document
Server: sDocServer
Database: sDocDb
Doc ID: NewID
☑ Render Into: "Body"

Then Goto: Add Additional Fields

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Source Server A script variable and the server of the database with the source document.

Source Database A script variable and the database with the source document.

Source Doc ID A script variable and the Notes ID of the source document.

Destination Server A script variable and the server of the database that stores the copied document.

Destination Database A script variable and the database name that will store the copied document.

Destination Doc ID Optional. The returned Notes ID of the copied document.

Render Into Contains the rich text version of the copied document.

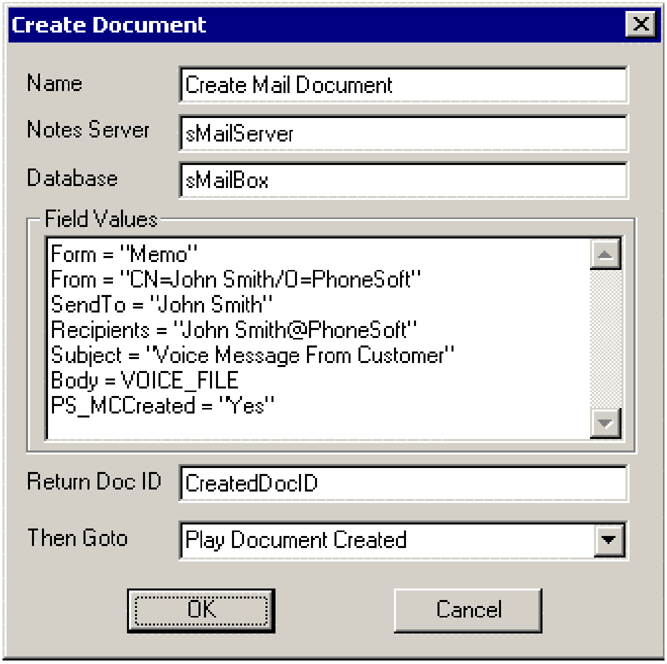
Then Goto The branch variable that is used after the command runs.

Create Document

Creates a document in the database specified. Each new field assignment statement has the following format:

Field_Name = Variable_Value

Where `Field_Name` is the name of a field and `Variable_Value` is the script variable in the new document.



Name: Create Mail Document

Notes Server: sMailServer

Database: sMailBox

Field Values:

```
Form = "Memo"
From = "CN=John Smith/O=PhoneSoft"
SendTo = "John Smith"
Recipients = "John Smith@PhoneSoft"
Subject = "Voice Message From Customer"
Body = VOICE_FILE
PS_MCCreated = "Yes"
```

Return Doc ID: CreatedDocID

Then Goto: Play Document Created

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Notes Server A script variable and the server name with the database that stores the new document.

Database A script variable and the database name that stores the document.

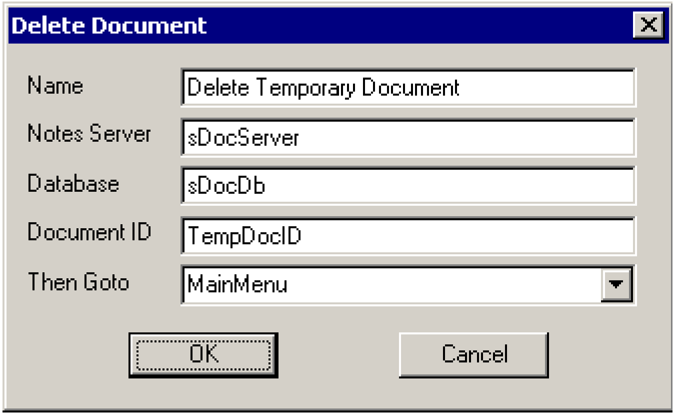
Field Values The fields and their corresponding values that are included in the new document.

Return DocID A script variable and the returned Notes ID of the new document. This variable is optional.

Then Goto The branch variable that is used after the command runs.

Delete Document

Deletes a document in the specified database.



Delete Document

Name: Delete Temporary Document

Notes Server: sDocServer

Database: sDocDb

Document ID: TempDocID

Then Goto: MainMenu

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Notes Server A script variable and the server name with the database that stores the document.

Database A script variable and the database name that stores the document.

Document ID A script variable and the Notes ID of the document.

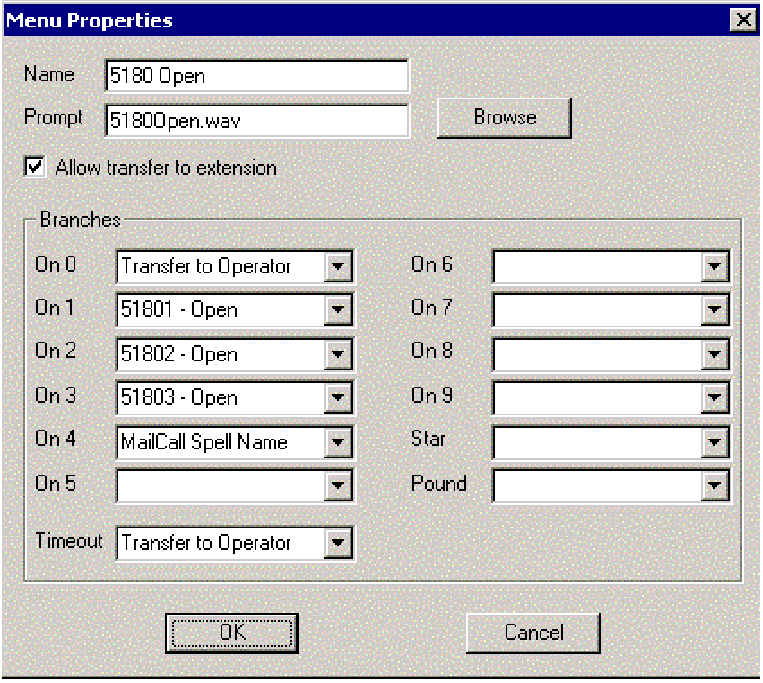
Then Goto The branch variable that is used after the command runs.

DTMF Menu

Plays a prompt and then uses the key entered by the caller to branch to the next specified command.

With this command, you can use the “Allow transfer to extension” field for setting the system to transfer a call to an extension.

The chosen menu option is assigned in the MENU_KEY reserved word.



Name: 5180 Open

Prompt: 5180Open.wav

☑ Allow transfer to extension

Branches

On 0	Transfer to Operator	On 6	
On 1	51801 - Open	On 7	
On 2	51802 - Open	On 8	
On 3	51803 - Open	On 9	
On 4	MailCall Spell Name	Star	
On 5		Pound	
Timeout	Transfer to Operator		

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

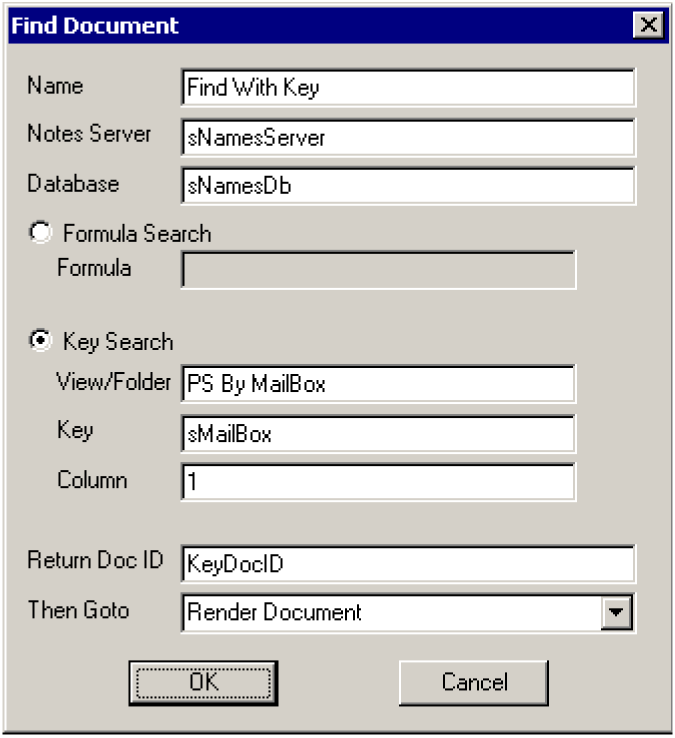
Prompt Plays while the system waits for the user to press a menu option. When the prompt is unnecessary, leave this field blank.

Allow transfer to extension When selected, a caller can enter a mailbox number. When a valid mailbox is entered, the call is transferred. When this field is not selected, the system uses and responds to the first number that the caller presses.

Branch On fields The branch variables that run after a caller presses corresponding number.

Find Document

Locates a document in the specified database. The two ways to search for a document are with a search formula or a search key.



Find Document

Name: Find With Key

Notes Server: sNamesServer

Database: sNamesDb

○ Formula Search

Formula:

● Key Search

View/Folder: PS By MailBox

Key: sMailBox

Column: 1

Return Doc ID: KeyDocID

Then Goto: Render Document

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Notes Server A script variable and the server name with the database that is searched.

Database A script variable and the database name that is searched.

Formula A script variable and the selection formula that is used for the search.

View/Folder A script variable and the view or folder that is used when searching with a database key lookup.

Key A script variable and the database key used in a search.

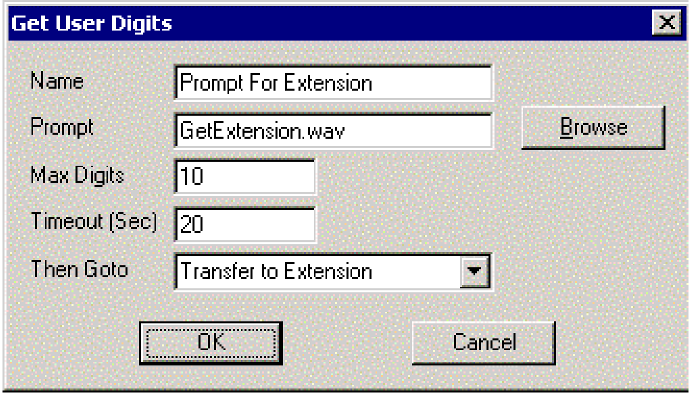
Column A script variable and the column name used with a database key lookup.

Return Doc ID A script variable and the Notes ID of the found document. If no document is found, 0 is the ID displayed.

Then Goto The branch variable that is used after the command runs.

Get User Digits

Captures the digits that the caller enters and logs the results in the DIGITS_STRING script variable. If there is a termination character, such as ✶ or #, the command inserts the character in the TERM_DIGIT script variable.



Name: Prompt For Extension
Prompt: GetExtension.wav
Max Digits: 10
Timeout (Sec): 20
Then Goto: Transfer to Extension

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Prompt The prompt that plays while waiting for the caller to enter a menu choice. If a prompt is unnecessary, leave this field blank.

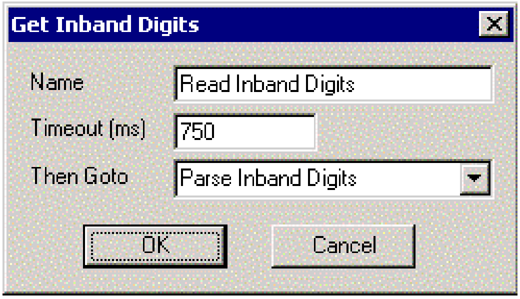
Max Digits Maximum number of digits that the caller can enter.

Timeout The number of seconds the caller has to enter the first digit.

Then Goto The branch variable that is used after the command runs.

Get Inband Digits

Captures inband digits from the phone system and logs the results in the INBAND_STRING script variable. Inband digits are those received from the phone system after the line has been picked up, but before the call is connected.



Get Inband Digits

Name: Read Inband Digits

Timeout (ms): 750

Then Goto: Parse Inband Digits

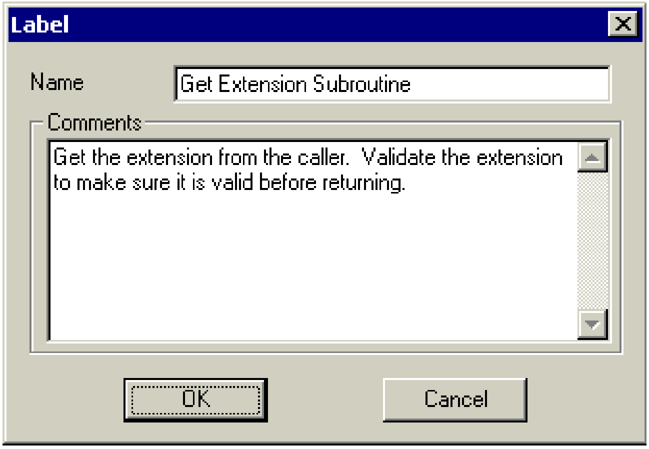
Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Timeout The number of milliseconds that the system waits for the phone system to send all of the inband digits.

Then Goto The branch variable that is used after the command runs.

Label

Tool for naming and describing a group of commands. This command requires no action by the system.



Name: Get Extension Subroutine

Comments: Get the extension from the caller. Validate the extension to make sure it is valid before returning.

Name Label name.

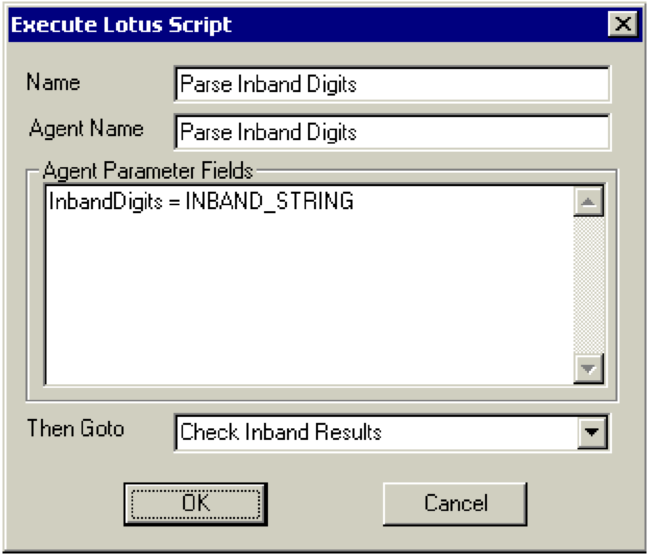
Comments Label description.

Execute LotusScript

Creates an in-memory Notes document and assigns the specified field values to the specified variable. Then the document is assigned to a Notes agent that resides in the database specified by the AGENT_SERVER and AGENT_DATABASE reserved words.

This Notes agent uses the specified fields to collect information from the script. If the agent is not a LotusScript agent, the field parameters are ignored. Use the following command line format to set up the Notes agent with the document context of the collected information:

```
NotesDocument = NotesSession.DocumentContext
```



Execute Lotus Script

Name: Parse Inband Digits

Agent Name: Parse Inband Digits

Agent Parameter Fields: InbandDigits = INBAND_STRING

Then Goto: Check Inband Results

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

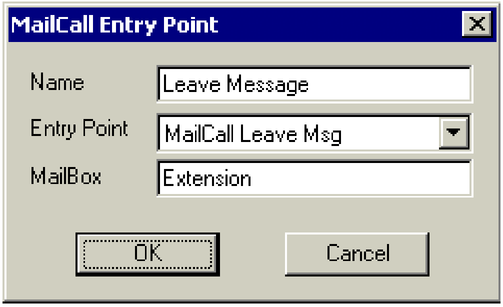
Agent Name The agent that runs.

Agent Parameter Fields Fields and their values that are passed to the in-memory Notes document.

Then Goto The branch variable that is used after the command runs.

MailCall Entry Point

Use to assign command actions at various points in the MailCall application.



MailCall Entry Point

Name: Leave Message

Entry Point: MailCall Leave Msg

MailBox: Extension

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Entry Point list

Use one of the following values to designate where to start the script:

MailCall Greeting The greeting that callers normally hear first.

MailCall Main Menu The main menu that callers hear after the company greeting; during which they can enter an extension or spell a name.

MailCall Spell Name This option gives callers access to the company phone extension directory to spell subscriber names.

MailCall Transfer When a mailbox is entered, the call is transferred to that mailbox. During the transfer, the caller hears “please wait while I transfer your call.” If an incorrect number is entered, the call is sent back to the main menu. This option performs a supervised or blind transfer based on the global Phone Server setting, which is different than the Phone Transfer command, on page 36.

MailCall Leave Msg When a mailbox is entered, the call is transferred to that mailbox and the caller hears the mailbox greeting. If an incorrect mailbox number is entered, the caller is asked to enter another mailbox number.

MailCall User Login A mailbox subscriber logs in and is prompted for a password. If an incorrect mailbox or password is entered, the subscriber is prompted to enter a valid one.

MailCall Goodbye Plays the goodbye prompt and then hangs up and waits for another call.

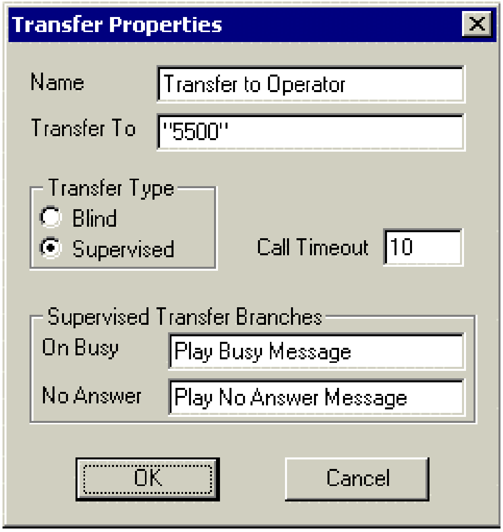
MailCall Hang-Up Hangs up and waits for another call without playing a goodbye prompt.

MailCall AMIS Receive Use for systems connected to other AMIS systems; set to wait for an AMIS tone while running the script.

MailBox A script variable. If the entry point allows the user to specify a mailbox, the transfer goes directly to the specified mailbox.

Phone Transfer

Transfers calls to the specified number. You can set up either a blind or a supervised transfer with this command.



Transfer Properties

Name: Transfer to Operator

Transfer To: "5500"

Transfer Type

○ Blind

● Supervised

Call Timeout: 10

Supervised Transfer Branches

On Busy: Play Busy Message

No Answer: Play No Answer Message

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Transfer To A script variable and the number where the call is to be transferred.

Transfer Type Set as a blind transfer or a supervised transfer. A blind transfer hangs up before the call transfer is complete. A supervised transfer waits to confirm the call is answered. If the call is unanswered or busy, then the call can be redirected as appropriate.

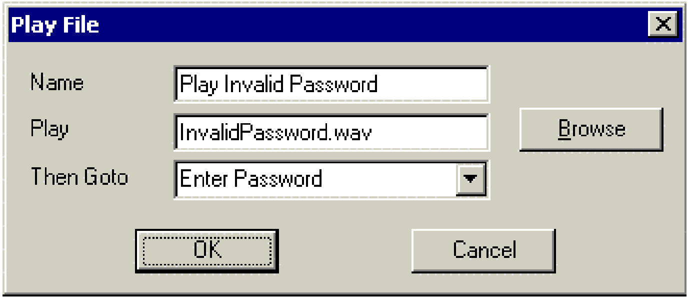
Call Timeout Use for a supervised call transfer; sets the amount of time the system waits before redirecting the call to the "No answer" command.

On Busy Runs when a call is busy during a supervised transfer.

No Answer Runs when a call is unanswered during a supervised transfer.

Play File

Plays a prompt to callers at the specified entry point.



Play File

Name: Play Invalid Password

Play: InvalidPassword.wav

Then Goto: Enter Password

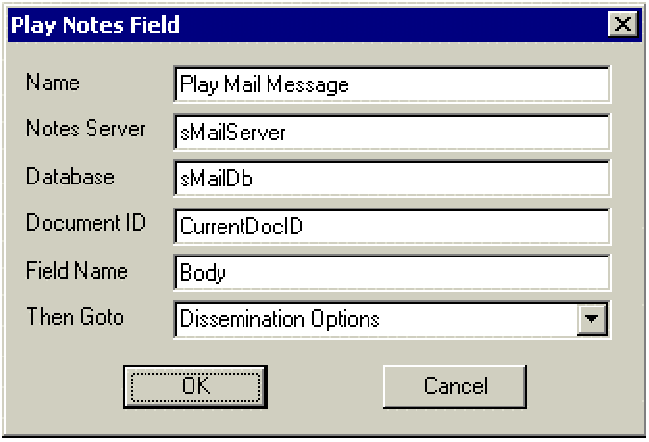
Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Play Can be a script variable and is the file name of the prompt that is played at the specified entry point.

Then Goto The branch variable that is used after the command runs.

Play Notes Field

Plays a WAV file or uses TTS (text to speech) to play text, rich text or numeric data that is contained in a Notes field.


Play Notes Field

Name: Play Mail Message

Notes Server: sMailServer

Database: sMailDb

Document ID: CurrentDocID

Field Name: Body

Then Goto: Dissemination Options

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Notes Server A script variable and the Notes server name with the database that stores the document.

Database A script variable and the database name that stores the document with the Notes field.

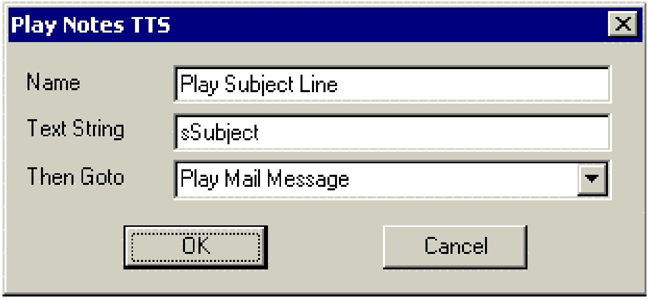
Document ID A script variable and the Notes ID of the document with the Notes field.

Field A script variable and the name of the Notes field, which can be text, rich text, or numeric.

Then Goto The branch variable that is used after the command runs.

Play Notes TTS

Plays the specified text string by using TTS.



Name: Play Subject Line

Text String: sSubject

Then Goto: Play Mail Message

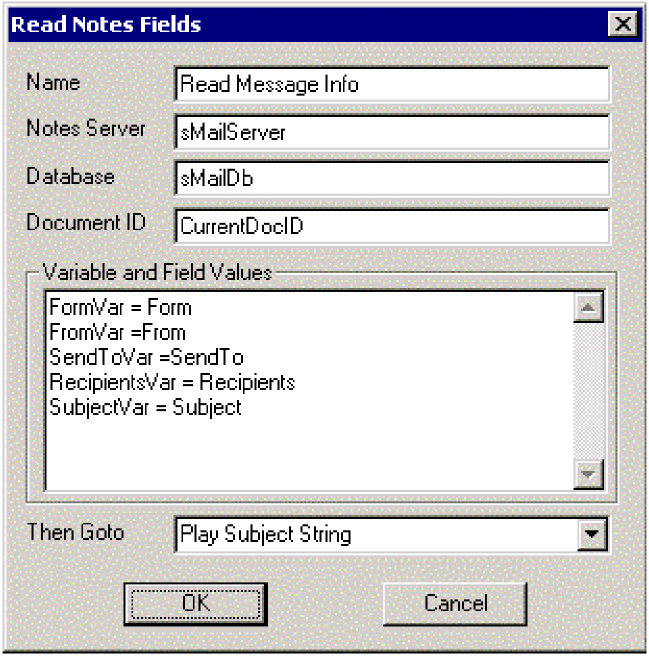
Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Text String A script variable and the Notes server name with the database that stores the Notes field.

Then Goto The branch variable that is used after the command runs.

Read Notes Fields

Reads multiple Notes fields and assigns them to local variables.



Read Notes Fields

Name: Read Message Info

Notes Server: sMailServer

Database: sMailDb

Document ID: CurrentDocID

Variable and Field Values:

```
FormVar = Form
FromVar =From
SendToVar =SendTo
RecipientsVar = Recipients
SubjectVar = Subject
```

Then Goto: Play Subject String

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Notes Server A script variable and the Notes server name with the database that stores the Notes fields.

Database A script variable and the database name that stores the Notes fields.

Document ID A script variable and the Notes ID of the document with the Notes fields.

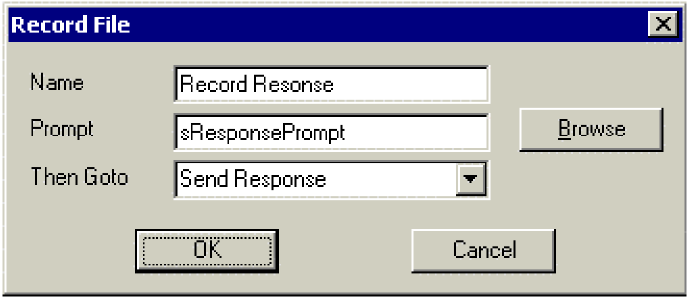
Variable and Field Values The list of assignment pairs. Use the following format; where "Notes_field" is a script variable:

```
<variable> = <Notes_field>
```

Then Goto The branch variable that directs the script after the command runs.

Record File

Records a voice message and stores the results in the VOICE_FILE script variable. Termination characters are stored in the TERM_DIGIT script variable.



Name: Record Resonse

Prompt: sResponsePrompt

Then Goto: Send Response

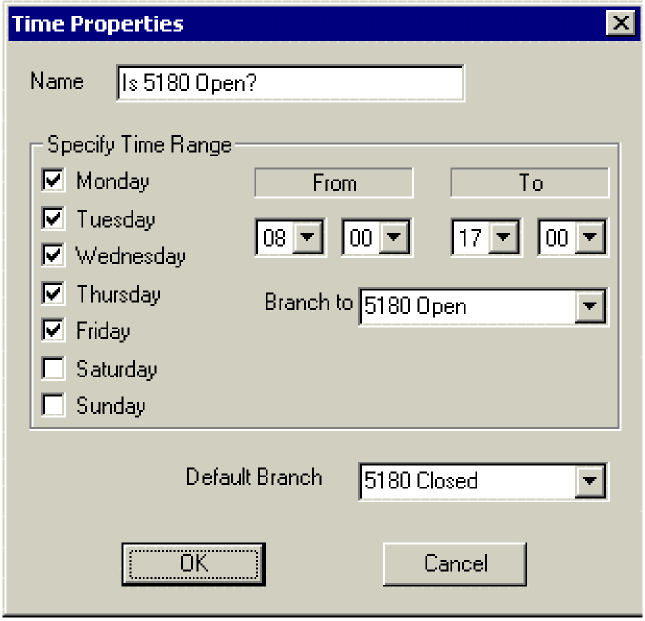
Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Prompt A script variable and the name of the prompt that a caller hears prior to recording a message.

Then Goto The branch variable that directs the script after the command runs.

Time Properties

If the current time is within the specified time range, this command branches to the command that is specified in the “Branch to” field.



Time Properties

Name: Is 5180 Open?

Specify Time Range

☑ Monday
☑ Tuesday
☑ Wednesday
☑ Thursday
☑ Friday
□ Saturday
□ Sunday

From: 08 : 00
To: 17 : 00

Branch to: 5180 Open

Default Branch: 5180 Closed

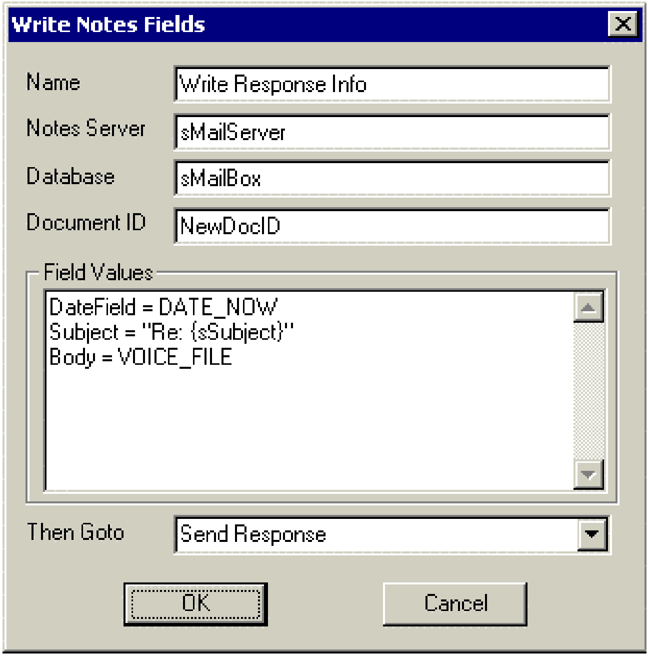
Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Time Range These fields specify the current time range that will result in a branch to a branch variable.

Default Branch The default is used when the current time is outside of the specified time range.

Write Notes Fields

Use to copy multiple Notes fields to a specified document.



Write Notes Fields

Name: Write Response Info

Notes Server: sMailServer

Database: sMailBox

Document ID: NewDocID

Field Values:

```
DateField = DATE_NOW
Subject = "Re: {sSubject}"
Body = VOICE_FILE
```

Then Goto: Send Response

Name Unique name for this command. This name is used by other commands to determine where to branch to next.

Notes Server A script variable and the server name with the database with the copied fields.

Database A script variable and the database name of the document with the copied fields.

Document ID A script variable and the Notes ID of the document with the copied fields.

Variable and Field Values The list of assignment pairs. Use the following format, where `<variable>` is a script variable:

Notes_Field = <variable>

Then Goto The branch variable that directs the script after the command runs.

Continued

Write Notes field examples

The following examples assume the fields do not already exist in the document. If a field already exists and you want to avoid adding values to the end of existing field values, delete the field before the copy by setting the field to `= empty`.

Before copying a voice file with this command, use the Record File command to store the voice file in the VOICE_FILE reserved word. Then you can set the field to `= VOICE_FILE`. If you want to append the voice file with any other value, you need to use LotusScript.

Assign Document Field as:	**Document Field Type results:**
Voice file	Does not assign the field. You must use LotusScript
Dialog list	Text List, in which `Item2 | 2` stores the value `2` (since an alias is used) and `Item2` stores as `Item2`.
Text list; set `field = “Text1`” and then `set field = “Text2`”	Text List stores as “Text1, Text2”
Text; set `field = “Text`”	Text List stores as “Text”
Date; set `field = DATE_NOW`	Date/Time List or Range stores as current date & time; you can append dates, but cannot append `DATE_NOW` with text.

CHAPTER 3: HelpDesk application

Using HelpDesk

With HelpDesk, you can send requests and messages to the Notes HelpDesk database by using either a phone or a computer with a microphone and speakers.

The demo is set up to automatically use file attachments. For the demo, you can use any sound player application. Confirm the computer is set up to automatically play a WAV file when opened. PhoneSoft supports both OLE (object linking and embedding) and file attachments for handling voice files.

PhoneSoft supports caller ID and ANI (automatic number identification). You can set up an application to collect the caller ID or ANI information for incoming calls. PhoneSoft also supports DID (direct inward dial) and DNIS (dialed number identification service), which can help multiple applications share a single phone line.

To open the demo database

1. Locate the Hdeskdat.nsf file. The installation default directory is C:\PhoneSoft\HelpDeskDemo\Hdeskdat.nsf.
2. Copy this file to your Notes Data directory.
3. From the Lotus Notes client main menu, click the File>Database>Open menus.
4. Locate and open the Hdeskdat.nsf file.
5. After opening the file, you can see how the database is set up.

   For example, select the By Status Number view. In the “Phone Message” and “Voice Status” fields, double-click the PhoneSoft voice recording icon and then click “Launch.”

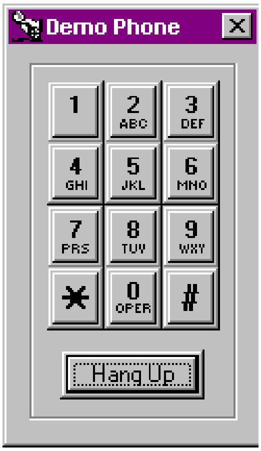
Starting the demo

The HelpDesk application is set up to automatically start when you start Phone Server. If the PhoneSoft computer is installed and set up with a voice board, you can call the system from any phone and the HelpDesk application answers the call.

You can also use the PhoneSoft Demo phone to run the application on any multimedia computer with a microphone and speakers; no voice board needed.

To simulate a call, confirm the application is running on phone line number one and then click the PhoneSoft Demo Phone smart icon. Your computer speakers play what a caller hears.

Click the Demo Phone keypad buttons to simulate a caller pressing numbers with a phone.



To submit a HelpDesk request

1. Click 1 to leave a help request.
2. When asked to enter an extension, click 2468#.
3. When prompted, record a message with the computer’s microphone and then click #.
4. When asked to enter a priority, click the appropriate number.
5. Write down the status number for your request. You will use it later to check your request status.
6. When you hear the opening greeting, click ✶ to hang up. A Help Request form is created in the Notes HelpDesk database.

Continued

To respond to the HelpDesk request

After the HelpDesk request is sent, a page with the status number is automatically sent to the HelpDesk administrator. The HelpDesk administrator can then respond to the new request:

1. To simulate the call, click the demo smart icon.
2. When asked to check the status of an existing request, press or click 3, which is a hidden menu selection for administrators.
3. When asked to enter the administration password, click 123#.
4. When asked to enter the status number of the request, click the status number that you wrote down in step 5 of the previous procedure and then click #.
5. You then hear the selected request message.
6. When asked to leave a response to this request, click 1 and then click 2 to continue without leaving a response.
7. Click 1 to leave a message for the person that sent the request.
8. When prompted, record a response with your computer microphone and then click #.

   HelpDesk stores the response in the “Response” field and sets a request complete flag in the “Help Request Form” document.

9. When asked to enter a status number, click ✶ to return to the main menu and click ✶ again to hang up.

To check for a request response

1. To simulate the call, click the demo smart icon.
2. When asked to check the status of an existing request, click 2.
3. When asked to enter the status number, click the number followed by #.
4. You then hear the administrator's response.
5. When you hear the opening greeting, click ✶ to hang up.

HelpDesk script

This section displays the HelpDesk script details and describes the values used for each script command. For field descriptions, see the corresponding command's topic in "Script commands" on page 19.

The script displays icons and names for each customized command. The plus and minus directory structure displays which commands are grouped together. The commands can be listed in any order, however these expanding levels help to organize the script.

To view a HelpDesk script command

1. Open the Script Generator. See "To start the Script Generator" on page 8.
2. Open the HelpDesk script. See "To open an existing script" on page 8.
3. In the HelpDesk script window, double-click the icon next to the command.

HelpDesk prompts

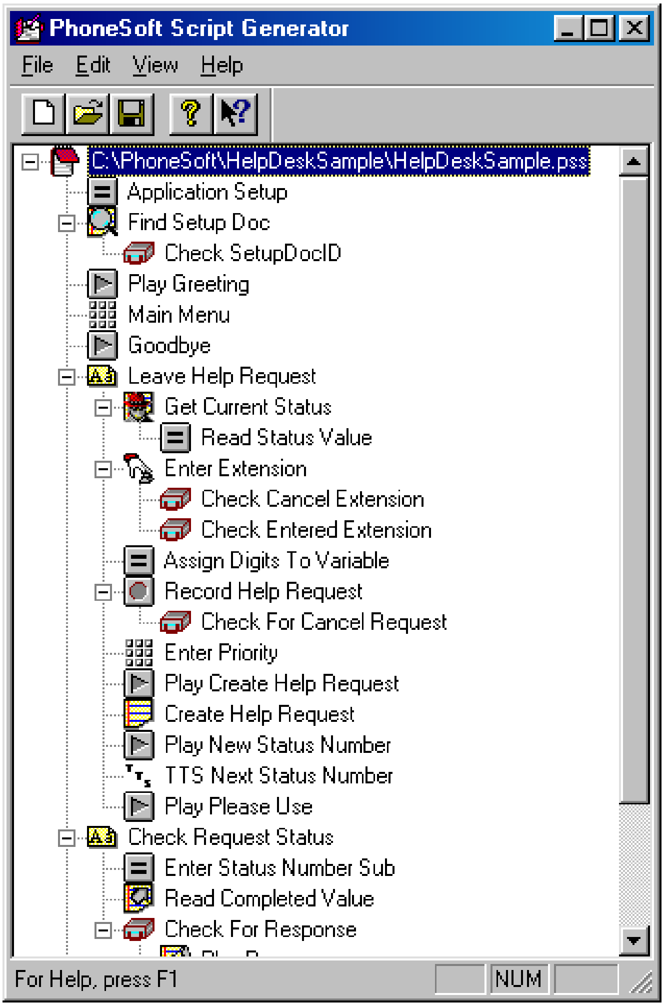
By default, the prompts included with the HelpDesk application are stored in the C:\PhoneSoft\Helpdeskdemo\Hdprompts file folder.

In the various script commands, ".\" begins the path in each "Play" field value. For example, in "Play Greeting" on page 56, the "Play" field value is ".\Hdprompts\Play_welcome.wav."

The ".\" included in the field value sets the application to look for the specified file in the folder of the currently running application. See also, ".\" in the Glossary on page 122.

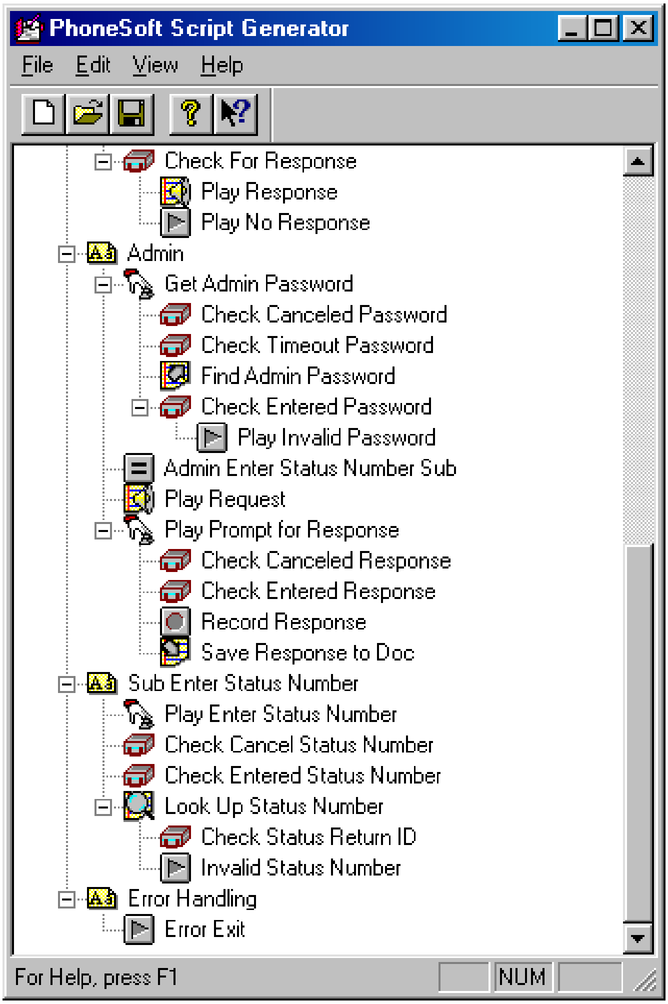
Continued

HelpDesk application script, page1



- C:\PhoneSoft\HelpDeskSample\HelpDeskSample.pss
  - Application Setup
  - Find Setup Doc
    - Check SetupDocID
  - Play Greeting
  - Main Menu
  - Goodbye
  - Leave Help Request
    - Get Current Status
      - Read Status Value
    - Enter Extension
      - Check Cancel Extension
      - Check Entered Extension
    - Assign Digits To Variable
    - Record Help Request
      - Check For Cancel Request
    - Enter Priority
    - Play Create Help Request
    - Create Help Request
    - Play New Status Number
    - TTS Next Status Number
    - Play Please Use
  - Check Request Status
    - Enter Status Number Sub
    - Read Completed Value
    - Check For Response

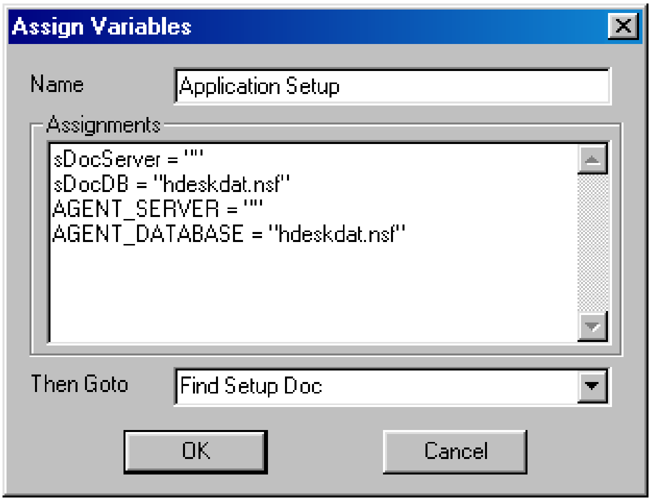
HelpDesk application script, page 2



```
    Check For Response
        Play Response
        Play No Response
Admin
    Get Admin Password
        Check Canceled Password
        Check Timeout Password
        Find Admin Password
        Check Entered Password
            Play Invalid Password
    Admin Enter Status Number Sub
    Play Request
    Play Prompt for Response
        Check Canceled Response
        Check Entered Response
        Record Response
        Save Response to Doc
Sub Enter Status Number
    Play Enter Status Number
    Check Cancel Status Number
    Check Entered Status Number
    Look Up Status Number
        Check Status Return ID
        Invalid Status Number
Error Handling
    Error Exit
```

Application Setup

The first command in the HelpDesk application defines where the HelpDesk database resides. HelpDesk uses this database to create and access the requests.



Name: Application Setup

Assignments

```
sDocServer = ""
sDocDB = "hdeskdat.nsf"
AGENT_SERVER = ""
AGENT_DATABASE = "hdeskdat.nsf"
```

Then Goto: Find Setup Doc

sDocServer = ““ A variable that defines the server of the database where the HelpDesk requests are created and accessed. Assigned as an empty string so that the local computer is used. To use a different computer, substitute the computer’s name, for example: `sDocServer = “Development/PhoneSoft”`

sDocDB = “HDeskDat.nsf” A variable that defines the target database.

To prevent changing all the paths when storing the data in differing directories, avoid hard coding paths in your application. To keep the database open while running the script, assign the path to the database name, which points the application to the Notes Data directory.

To use a different path, include `.\` and the application locates the specified file in the directory of the currently running application. For example:
`sDocDB = “.\HDeskDat.nsf”`

If you use this setting, the database cannot be open while running the script that is using the database.

NOTE: *Define reserved words separately and do not change the variable name.*

AGENT_SERVER= “” A reserved word that defines where the Lotus agent resides.

AGENT_DATABASE = “HDeskDat.nsf” A reserved word that is the name of the database that contains the agent. Because the database name is used in the path assignment, the path of the database is assumed by Phone Server to be in the Notes Data directory.

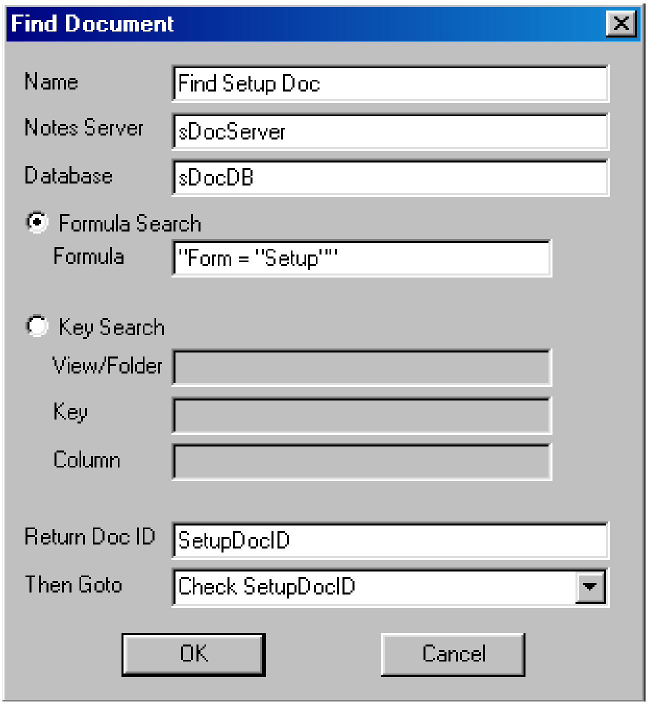
Then GoTo Branches to “Find Setup Doc” after this command runs.

Find Setup Doc

HelpDesk uses a database, “HDeskDat.nsf” to store and update help requests and responses to the requests.

With “Find Setup Doc,” the system confirms that the database exists by locating the database with the Setup document. Then the script assigns the document an ID.

NOTE: *To manually find the Setup document, open “HDeskDat.nsf” and click the Setup Doc view. Locate the HelpDesk Setup document, which stores the “Next status number” and “System password” fields.*



Find Document

Name: Find Setup Doc

Notes Server: sDocServer

Database: sDocDB

● Formula Search

Formula: "Form = "Setup""

○ Key Search

View/Folder:

Key:

Column:

Return Doc ID: SetupDocID

Then Goto: Check SetupDocID

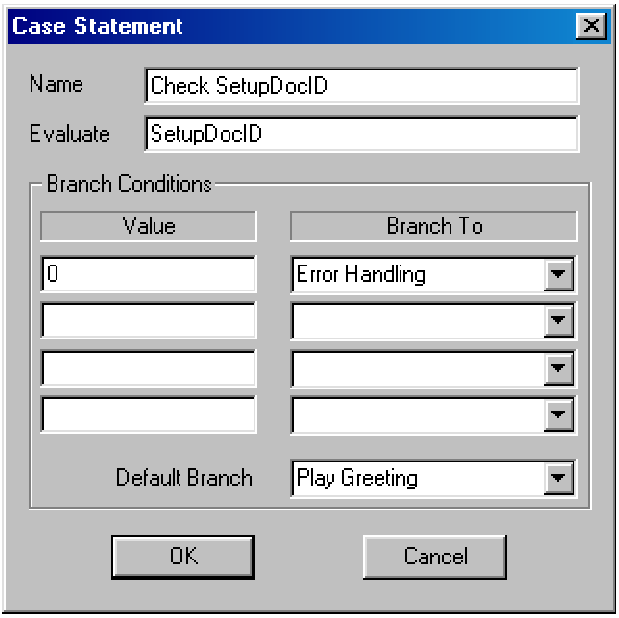
sDocServer and sDocDB Variables defined in “Application Setup.”

“Form = “Setup”” Searches for the Setup form.

“SetupDocID” Variable name assigned to the document found. This variable is used to reference this document in the rest of the application.

Check SetupDocID

Defines an error message if the document does not exist.



Name: Check SetupDocID

Evaluate: SetupDocID

Branch Conditions

Value	Branch To
0	Error Handling

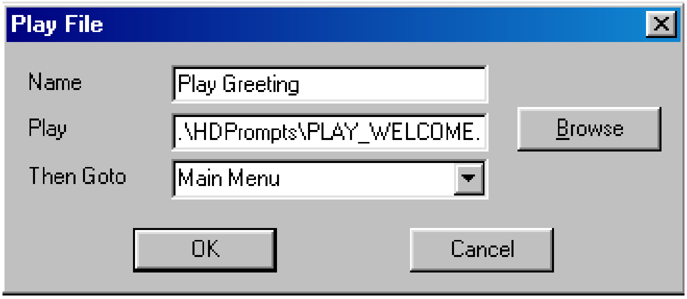
Default Branch: Play Greeting

Evaluate The Case Statement evaluates the SetupDocID.

Branch Conditions IIf the document fails to exist, the value returned from evaluating SetupDocID is 0 and the script branches to "Error handling," see page 106. Any other return values represent an existing document and the script branches to "Play Greeting," see page 56.

Play Greeting

Branch variable that plays a recorded voice file, is stored in the HelpDeskDemo directory.



Play File

Name: Play Greeting

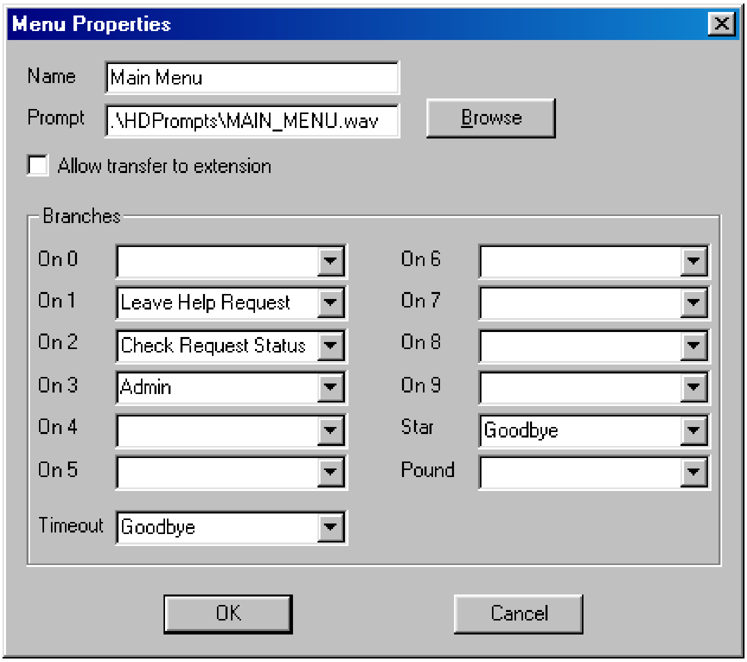
Play: .\HDPrompts\PLAY_WELCOME.

Then Goto: Main Menu

.\Hdprompts\Play_welcome.wav The location of the recording, which is a voice file in the Hdprompts folder in C:\PhoneSoft\HelpDeskDemo.

Main Menu

HelpDesk branches to “Main Menu” after the caller hears: “Welcome to the PhoneSoft HelpDesk. Press 1 to leave a help request. Press 2 to check on the status of an existing request.”



Menu Properties

Name Main Menu

Prompt .\HDPrompts\MAIN_MENU.wav

□ Allow transfer to extension

Branches

On 0		On 6	
On 1	Leave Help Request	On 7	
On 2	Check Request Status	On 8	
On 3	Admin	On 9	
On 4		Star	Goodbye
On 5		Pound	
Timeout	Goodbye		

.\Hdorompts\Main_menu.wav The location of the recording, which is a voice file in the Hdprompts folder in C:\PhoneSoft\HelpDeskDemo.

On 1 If the caller presses 1, the script branches to the Leave Help Request command.

On 2 Branches to the Check Request Status command.

On 3 Branches to the Admin command.

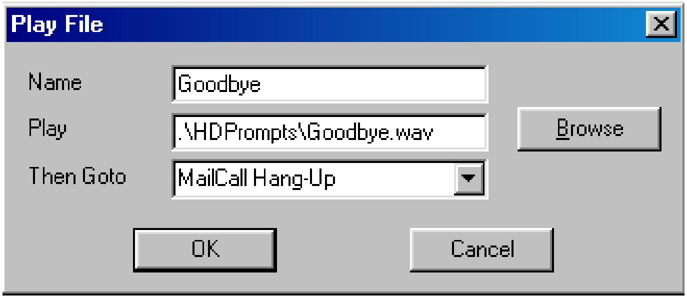
Star If the caller presses ✶, it branches to the Goodbye command.

Timeout If the caller fails to press anything, the script branches to the Goodbye command.

Other fields left blank Send the caller back to “Main Menu.”

Goodbye

HelpDesk hangs up after playing the goodbye message, “Thank you for calling. Goodbye.”



Name: Goodbye

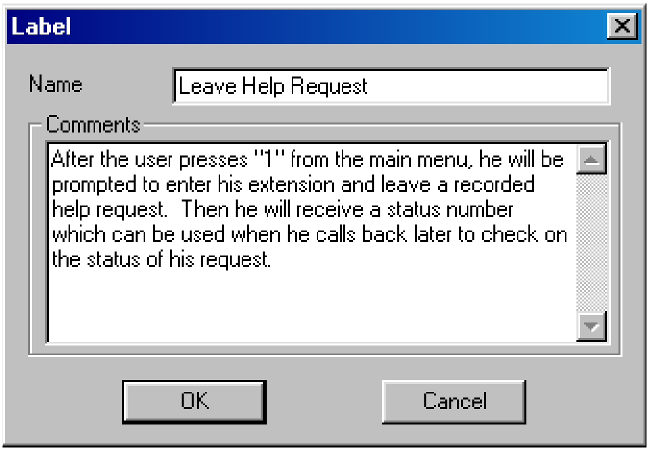
Play: .\HDPrompts\Goodbye.wav

Then Goto: MailCall Hang-Up

.\Hdprompts\Goodbye.wav The location of the prompt.

Leave Help Request

Label command that groups related commands.



Leave Help Request

After the user presses "1" from the main menu, he will be prompted to enter his extension and leave a recorded help request. Then he will receive a status number which can be used when he calls back later to check on the status of his request.

Comments Stores comments about this group of commands. For reference purposes only, no action is taken.

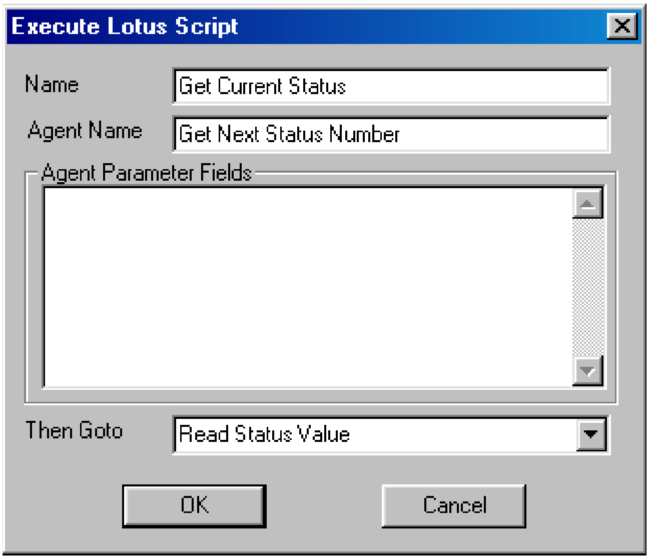
NOTE: *A "Then GoTo" field is unnecessary for this command. The application branches to the next command listed, which for HelpDesk is "Get Current Status."*

Get Current Status

Uses a Lotus agent, which is programmed in LotusScript. For details, see "Get Next Status Number" on page 108.

The Lotus agent locates the number in the "NextStatus" field and stores that value in the "NextStatus" variable. The agent then increments the status number in the HelpDesk document. The "NextStatus" variable is then used in the next command, "Read Status Value."

For example, if the current status in the Setup document is "100," the agent assigns "100" to the "NextStatus" variable. Then the agent increments the "NextStatus" field to use "101" for the next request.



Name: Get Current Status

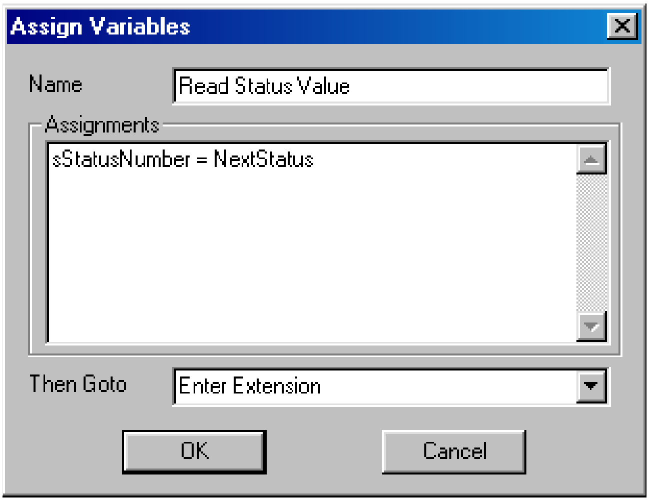
Agent Name: Get Next Status Number

Then Goto: Read Status Value

Get Next Status Number The agent that is created in the HDeskDat.nsf file.

Read Status Value

Assigns the value in the “NextStatus” field to the “sStatusNumber” variable.



Assign Variables

Name: Read Status Value

Assignments:

```
sStatusNumber = NextStatus
```

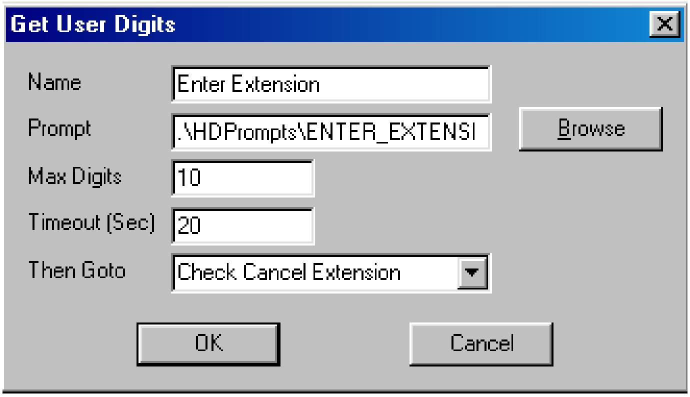
Then Goto: Enter Extension

Assignments “StatusNumber = NextStatus” uses the “sStatusNumber” variable to reference the current status in the Setup document. The “NextStatus” variable is used by a Lotus agent and stores the status number that the agent looks up in the Setup document.

Enter Extension

The application is still gathering information for the help request document. Variables store the information before the script creates the document. Without the variables, the document has no information to store.

“Enter Extension” is a prompt that asks callers to enter their extensions. The entered extension is stored in the “RequestedBy” and “Phone” fields of the Help Request document. The DIGITS_STRING reserved word stores the extension.


Get User Digits

Name: Enter Extension

Prompt: .\HDPrompts\ENTER_EXTENSI

Max Digits: 10

Timeout (Sec): 20

Then Goto: Check Cancel Extension

.\Hdprompts\Enter_extension.wav The location of the prompt.

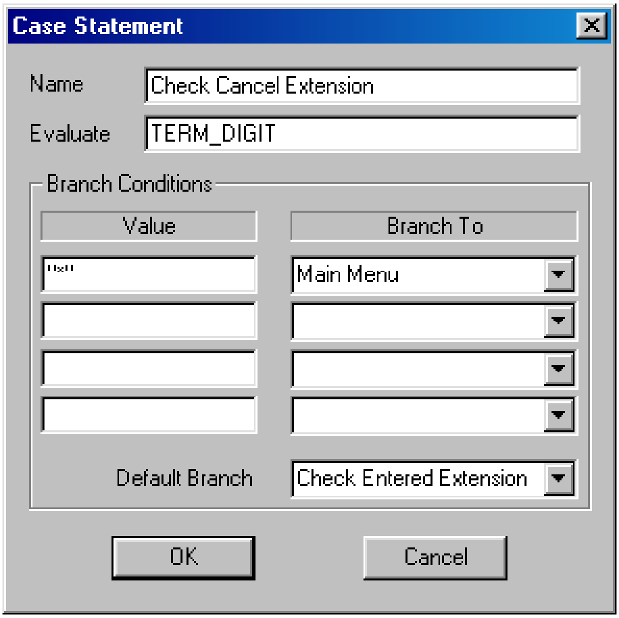
Max Digits 10 is the maximum number of digits a caller can enter.

Timeout (Sec) 20 seconds limit for a caller to enter the first digit.

Check Cancel Extension

After a caller enters an extension followed by #, the script stops the 20-second timeout and branches to the next command. You can set ✶ to wherever you want it to branch. Most automated systems and this application use ✶ to go back a level. With this command, ✶ sends the call back to “Main Menu” and cancels “Enter Extension.”

“Check Cancel Extension” checks the TERM_DIGIT reserved word for # and determines where to branch next.



Case Statement

Name: Check Cancel Extension

Evaluate: TERM_DIGIT

Branch Conditions

Value	Branch To
"*"	Main Menu

Default Branch: Check Entered Extension

TERM_DIGIT A reserved word used in the Get Digits or Record File commands. Termination Digits are either ✶ or #.

Branch Conditions ✶ is a termination character that returns the call to “Main Menu.” The value must be enclosed in double quotes, because it is a character string.

Default Branch Sends the script to “Check Entered Extension” if ✶ is not entered.

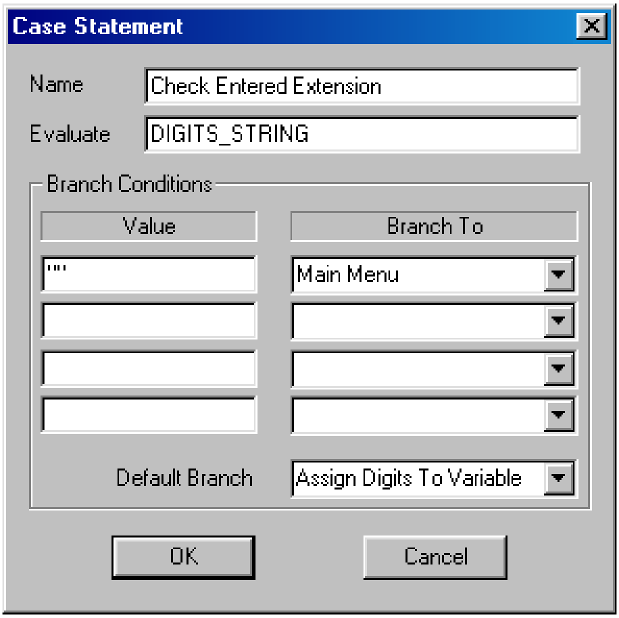
Check Entered Extension

Checks whether the caller hangs up or waits too long to enter a digit.

To avoid an infinite call loop keeping a phone line busy, the application needs to branch correctly when a caller hangs up. For example, if the application branches to “Enter Extension” when no digit is entered and the caller hangs up, no extension is ever entered and causes an infinite call loop.

NOTE: *To stop an infinite call loop, you must shut down Phone Server.*

To avoid the endless call, test the application by simulating a call and waiting without entering any digits. Eventually the application needs to branch to “Goodbye” and hang up. Or if you enter a valid digit at any time during the loop, the application needs to branch correctly.



Name: Check Entered Extension

Evaluate: DIGITS_STRING

Branch Conditions

Value	Branch To
""	Main Menu

Default Branch: Assign Digits To Variable

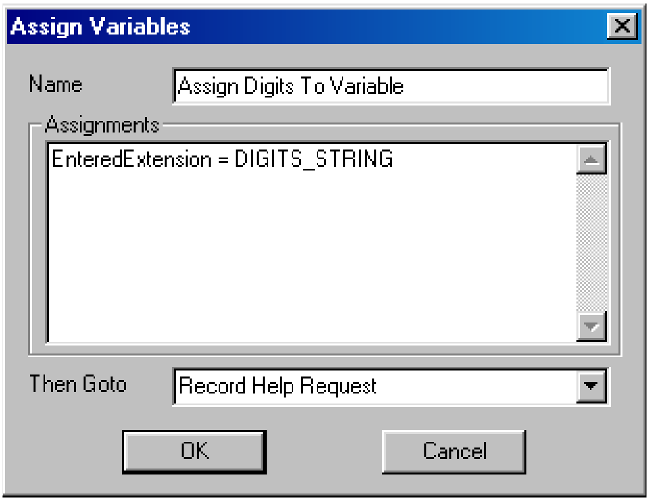
DIGITS_STRING A reserved word resulting from the Get Digits command.

Branch Conditions If nothing is entered, the value of DIGITS_STRING is null or empty. The two double quotes equals an empty string and signals the application to send the call back to the main menu.

Assign Digits To Variable

If a caller enters an extension, this command assigns the digits to a variable so the application can use the digits later. After the entered extension is stored in a variable, the application is ready to insert that extension in the request document.

Avoid using another Get Digits command before assigning the entered digits to a variable or any new digits entered overwrite the old digits in the DIGITS_STRING reserved word. Then the application can use DIGITS_STRING again for a new set of digits.



Name: Assign Digits To Variable

EnteredExtension = DIGITS_STRING

Then Goto: Record Help Request

EnteredExtension = DIGITS_STRING Assigns the digits in the DIGITS_STRING reserved word to the “EnteredExtension” variable.

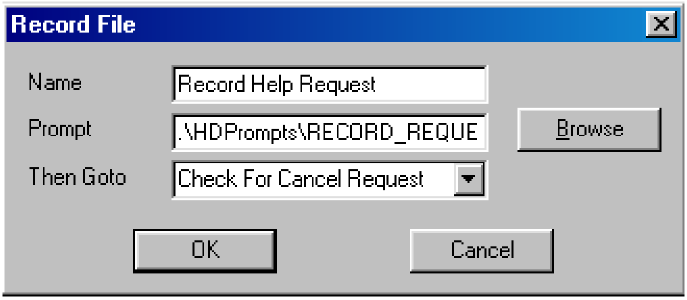
Record Help Request

Plays a prompt that asks the caller to record a help request after the tone.

***NOTE:** When using the demo phone, you do not hear a tone at the end of the prompt. However, a caller that uses a phone line does hear the tone.*

The recorded request is stored as a temporary WAV file in the VOICE_FILE script variable. This file is stored in the HelpDesk Temp directory, C:\Phone-Soft\Helpdeskdemo\Temp.

For example, a request at 11:06:55 AM on line 1 is recorded as an 1106551.wav file.



Record File

Name: Record Help Request

Prompt: .\HDPrompts\RECORD_REQUE

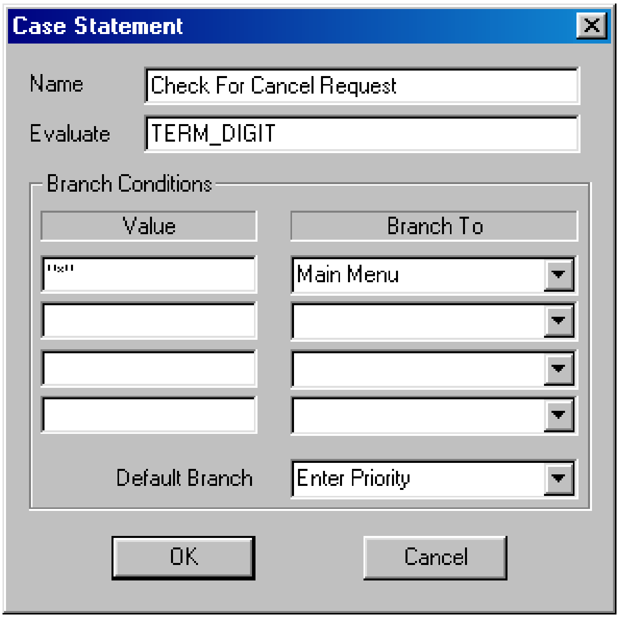
Then Goto: Check For Cancel Request

.\Hdprompts\Record_request.wav Location of the prompt.

Check For Cancel Request

Just as we check for a canceled extension earlier, the TERM_DIGIT reserved word checks for a canceled recording.

If the caller enters ✶ to cancel the recording, the application sends the call back to the main menu. If not, the application branches to “Enter Priority.”



Name: Check For Cancel Request

Evaluate: TERM_DIGIT

Branch Conditions

Value	Branch To
"*"	Main Menu

Default Branch: Enter Priority

TERM_DIGIT A reserved word used by the Get Digits or Record File commands. Termination digits are either ✶ or #.

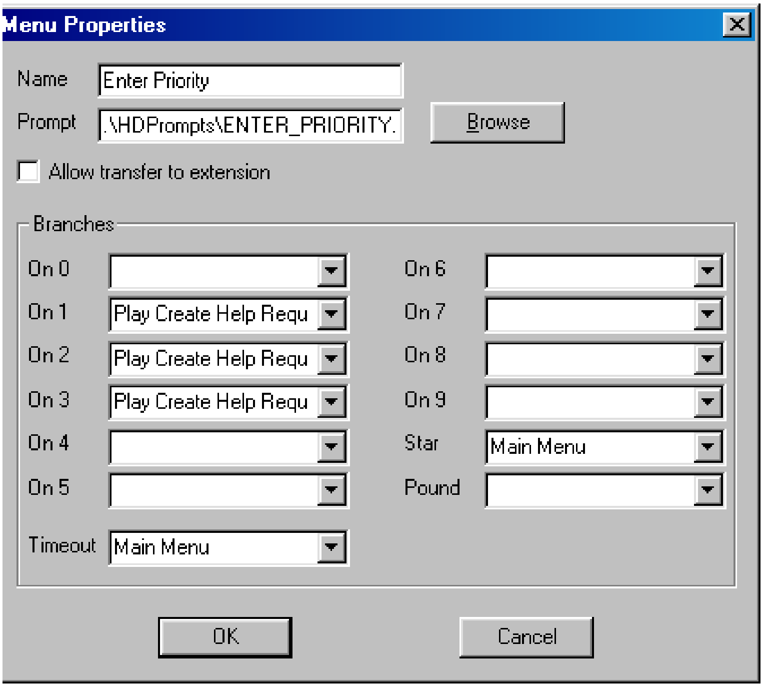
Branch Conditions ✶ sends a call to the main menu. The value is enclosed in double quotes because it is seen as a character string.

Enter Priority

After a caller records a help request, this command prompts the caller to enter the priority of the request; one for high, two for medium, or three for low. The entered priority is stored and then added to the help request document later.

The DTMF Menu command uses a MENU_KEY reserved word to store the entered priority.

You can use a Case Statement command, which requires more commands to check for a timeout or canceled action separately. By using a Menu Properties command, both actions are checked with the script command.



Name: Enter Priority

Prompt: .\HDPrompts\ENTER_PRIORITY.

☐ Allow transfer to extension

Branches

On 0		On 6	
On 1	Play Create Help Requ	On 7	
On 2	Play Create Help Requ	On 8	
On 3	Play Create Help Requ	On 9	
On 4		Star	Main Menu
On 5		Pound	

Timeout: Main Menu

.\Hdprompts\Main_menu.wav The location of the prompt.

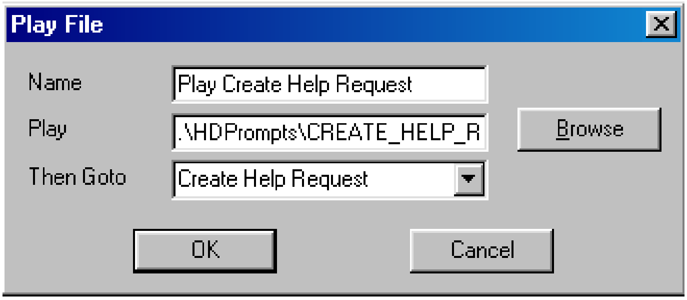
On 1, 2 or 3 Branches to “Play Create Help Request.”

Star (✶) or Timeout Branches to “Main Menu.”

Other branches Branches back to “Enter Priority” and repeats the prompt. Be cautious of an infinite call loop when using this value.

Play Create Help Request

Prompts the caller to wait while the request is added to the database.



Name: Play Create Help Request

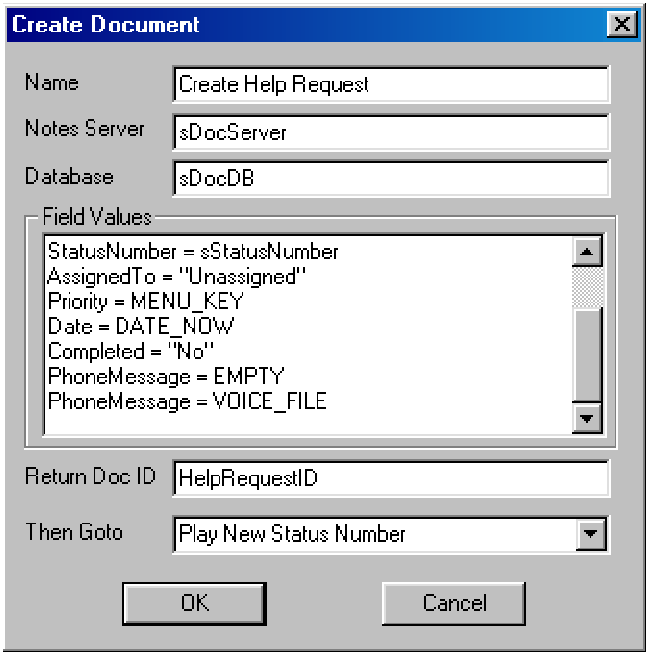
Play: .\HDPrompts\CREATE_HELP_R

Then Goto: Create Help Request

.\Hdprompts\Create_help_request.wav The location of the prompt.

Create Help Request

After all the information is collected, this command creates the help request.



Create Document

Name: Create Help Request

Notes Server: sDocServer

Database: sDocDB

Field Values:

```
StatusNumber = sStatusNumber
AssignedTo = "Unassigned"
Priority = MENU_KEY
Date = DATE_NOW
Completed = "No"
PhoneMessage = EMPTY
PhoneMessage = VOICE_FILE
```

Return Doc ID: HelpRequestID

Then Goto: Play New Status Number

sDocServer The server with the database that stores the requests. This variable is defined in “Application Setup.”

sDocDB The database that stores the requests. This variable is defined in “Application Setup.”

Field values The document field names are left of = (equal sign) and the field values are right of = (equal sign).

Extension, RequestedBy and StatusNumber Inserts the data that is stored in the cooresponding variables as field values in the document.

Help Request Name of the form used to create the document. All of the field names listed are fields in this form.

AssignedTo = “Unassigned” Inserts an unassigned value into the “AssignedTo” field, because the request is unassigned until later.

Priority = MENU_KEY Inserts the value from the “Enter Priority” branch into the “Priority” field of the request document.

Date = DATE_NOW Uses the DATE_NOW reserved word to insert the current date into the “Date” field.

Completed = “No” Set the field to “No.”

PhoneMessage = EMPTY Uses the EMPTY reserved word to clear the “PhoneMessage” Notes field. This prevents the new field value from being added to the end of an old value.

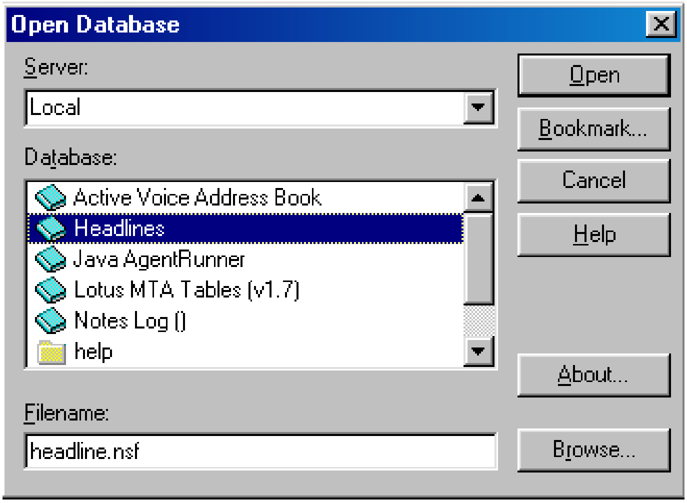
PhoneMessage = VOICE_FILE Uses the Record File command to insert the request recording into the “PhoneMessage” field.

Viewing help requests

Perform the following two procedures to open the HelpDesk database and view the new HelpDesk request.

To open the HelpDesk database

1. Open Notes or Domino and click the File>Database>Open menus.
2. In the Open Database dialog box, locate and select "HelpDesk Data."
3. At the bottom of the dialog box, confirm that the Hdeskdat.nsf file is displayed in the "Filename" field.



Open Database

Server: Local

Database:

- Active Voice Address Book
- Headlines
- Java AgentRunner
- Lotus MTA Tables (v1.7)
- Notes Log ()
- help

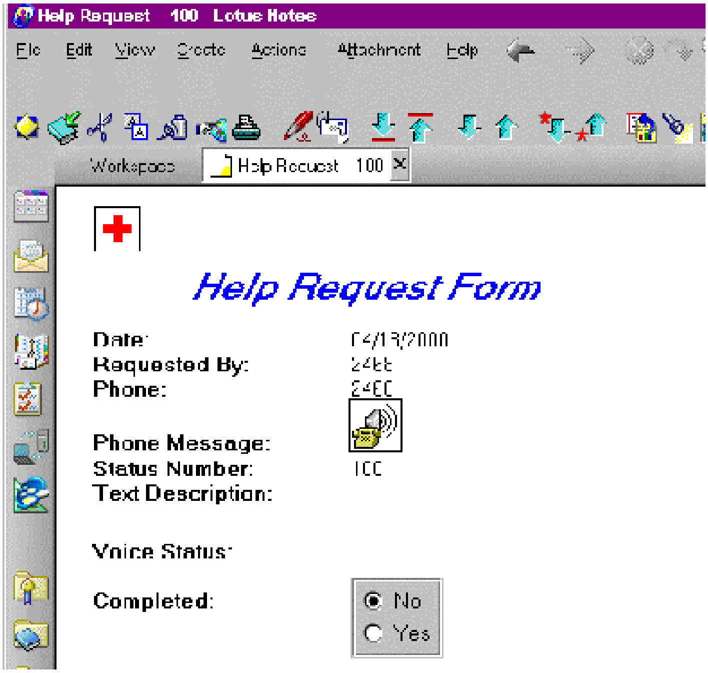
Filename: headline.nsf

4. Click "Open" and when the "Help Desk Data on Local" icon appears on your desktop, double-click it to open the database.

   CAUTION: *Because Notes allows multiple copies and replicas of the same database, confirm that you have the correct database open before making any changes.*

To open a help request document

1. In the “All” folder section of Views, click the “By Date” view.
2. In the right window pane, double-click the new help request.



Help Request Form

Date: 04/13/2000
Requested By: 2466
Phone: 2460
Phone Message:
Status Number: 100
Text Description:

Voice Status:

Completed: ● No ○ Yes

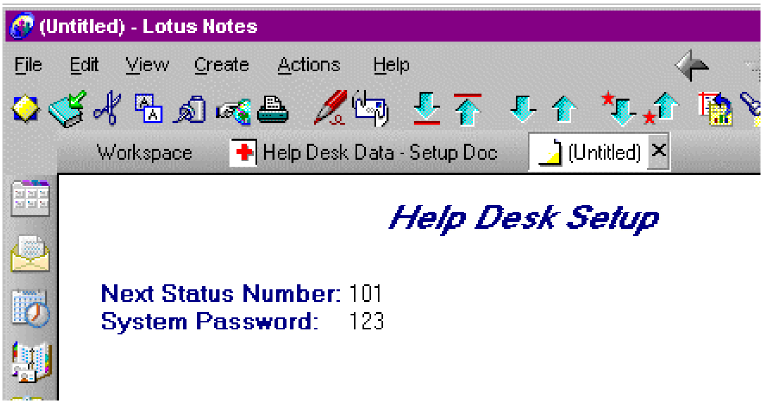
NOTE: After a request is submitted, look at the “Setup Doc” view to see how the Setup document changes and how the Status Number value changes from “100” to “101.”

Changing administrator password

Perform the following to change the administrator password for the HelpDesk application.

To change the password

1. Open a help request document.
2. Open the “Setup Doc” view.



Help Desk Setup

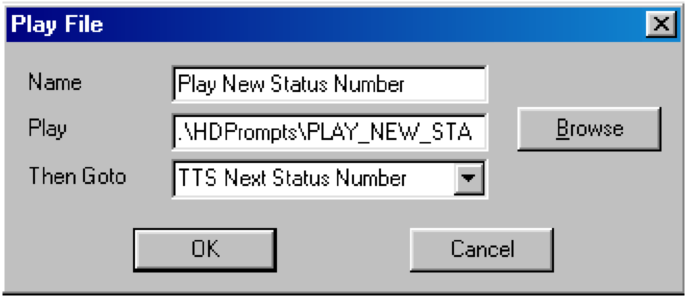
Next Status Number: 101
System Password: 123

3. Double-click the System Password field value.
4. Type in your new password and then click the File>Save menus.

Play New Status Number

After the request document is created, the caller needs to receive the the status number for checking the status of the request.

With this command, the caller hears “The status number that has been assigned to your request is” and the appropriate number. The next branch uses TTS to play the number.



Name: Play New Status Number

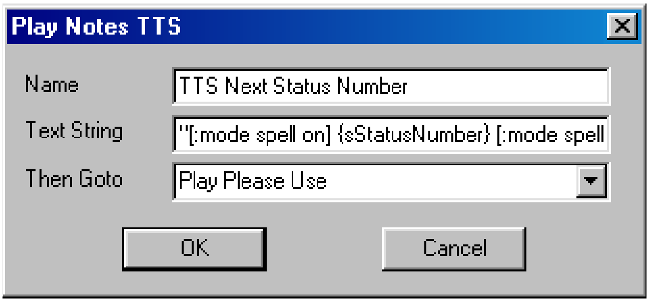
Play: .\HDPrompts\PLAY_NEW_STA

Then Goto: TTS Next Status Number

.\Hdprompts\Play_new_status_number.wav The location of the prompt.

TTS Next Status Number

Uses TTS to play the status number for the caller.



Play Notes TTS

Name: TTS Next Status Number

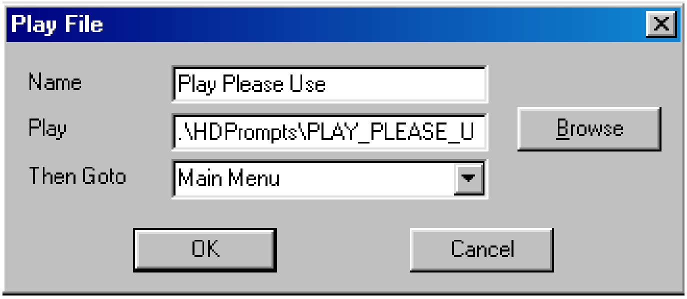
Text String: "[:mode spell on] {sStatusNumber} [:mode spell

Then Goto: Play Please Use

"[:mode spell on] {sStatusNumber} [:mode spell off]" Without the mode settings, the "sStatusNumber" variable uses TTS to read the whole number instead of reading the individual digits. For example, 100 is played as "one hundred" instead of "one zero zero." The "[:mode spell on]" changes TTS to read each digit and then "[:mode spell off]" sets the variable back.

Play Please Use

Prompts callers to “please use this number when calling in to check on the status of this request.” The application then transfers the call to the main menu.



Name: Play Please Use

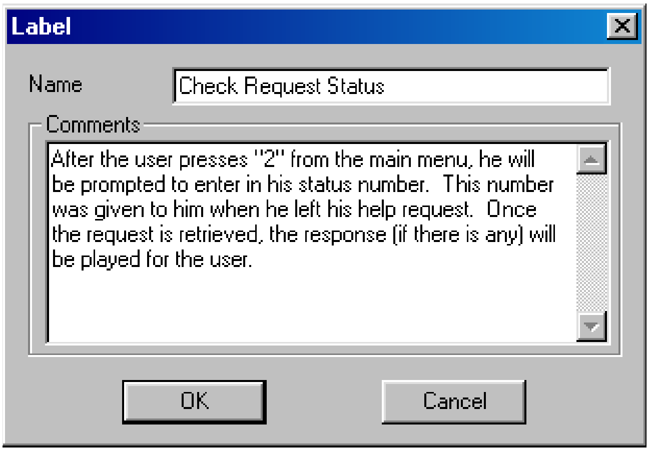
Play: .\HDPrompts\PLAY_PLEASE_U

Then Goto: Main Menu

.\Hdprompts\Play_please_use.wav The path and file name of the prompt.

Check Request Status

Label command that groups related commands. This group sets up the request status.



Name: Check Request Status

Comments: After the user presses "2" from the main menu, he will be prompted to enter in his status number. This number was given to him when he left his help request. Once the request is retrieved, the response (if there is any) will be played for the user.

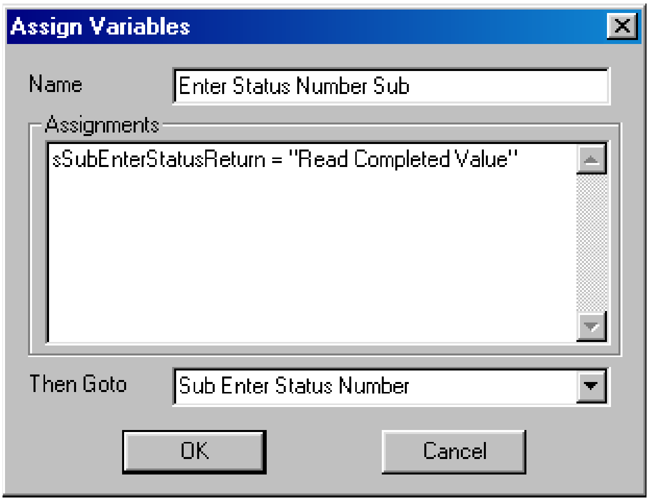
Comments Stores comments about this group or section of the script. For reference purposes only, no action is taken.

NOTE: *A "Then GoTo" field is unnecessary for this command. The application branches to the next command listed.*

Enter Status Number Sub

After a request is submitted, the next task for the caller is to come back and check on the progress of the request.

The “Sub Enter Status Number” subroutine branches to the “sSubEnterStatusReturn” variable. After the script runs all of the commands in that variable, this assignment sets the last action of the variable to go to the “Read Completed Value” branch.



Name: Enter Status Number Sub

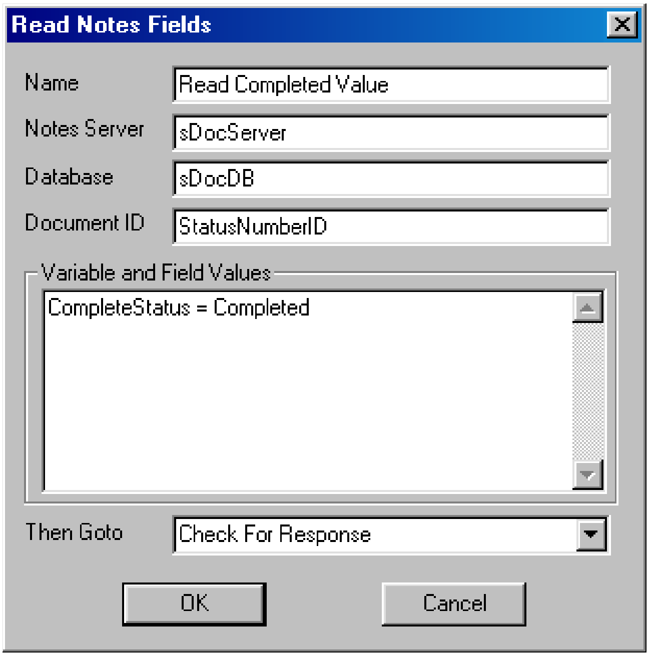
Assignments: sSubEnterStatusReturn = "Read Completed Value"

Then Goto: Sub Enter Status Number

sSubEnterStatusReturn = “Read Completed Value” Assigns the “Read Completed Value” to the “sSubEnterStatusReturn” variable. This assignment branches to “Read Completed Value” after the “Sub Enter Status Number” subroutine finishes.

Read Completed Value

Checks that the "Completed" field value is "Yes" by assigning the "Completed" field value to the "CompletedStatus" variable.



Read Notes Fields

Name: Read Completed Value

Notes Server: sDocServer

Database: sDocDB

Document ID: StatusNumberID

Variable and Field Values: CompleteStatus = Completed

Then Goto: Check For Response

sDocServer The name of the Notes Server of the target database. This variable is defined in "Application Setup."

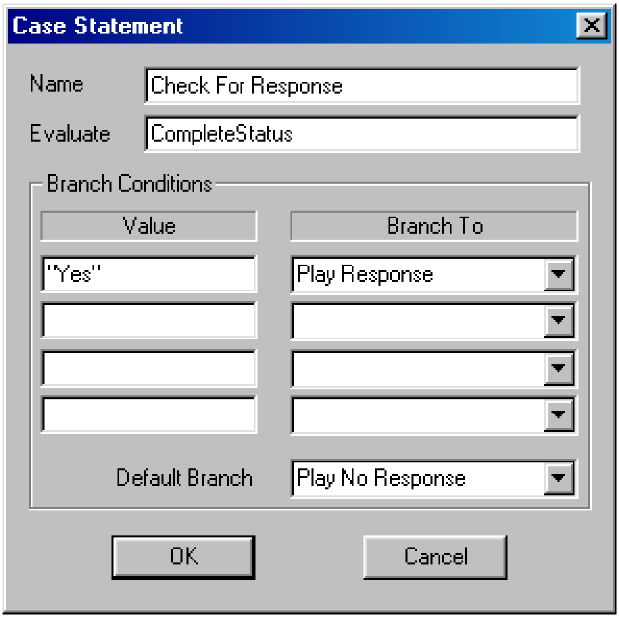
sDocDB The target database that stores the requests. This variable is defined in "Application Setup."

StatusNumberID Identifies the current document; this ID was chosen in the "Look Up Status Number" branch.

CompleteStatus = Completed "Completed" is the field with the value that the "CompleteStatus" variable uses. If the help request is completed, the "Completed" field is set to "Yes," which then sets the "CompletedStatus" value to "Yes."

Check For Response

Checks the value for the “CompletedStatus” variable. If it is set to “Yes,” the script branches to “Play Response.” If not, the script branches to “Play No Response.”



Case Statement

Name: Check For Response

Evaluate: CompleteStatus

Branch Conditions

Value	Branch To
"Yes"	Play Response

Default Branch: Play No Response

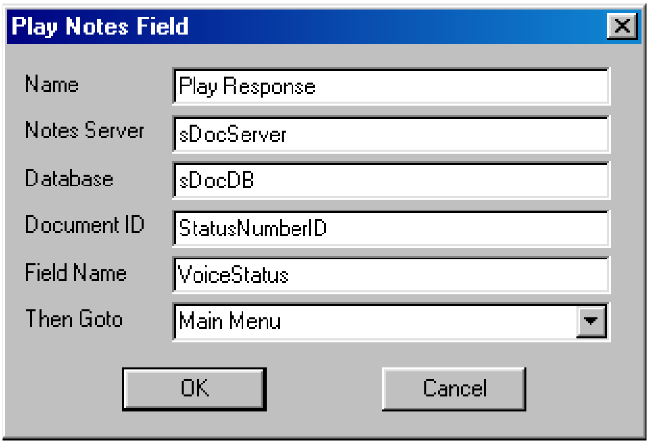
CompleteStatus A variable that defines the last branch depending on the “Completed” field value.

Branch Conditions If the field value is “Yes,” branch to “Play Response.” If the value does not equal “Yes,” branch to “Play No Response.”

Play Response

Plays the contents of the VocieStatus field.

NOTE: *Only when a response is left, is the "Completed" field value set to "Yes."*



Name: Play Response

Notes Server: sDocServer

Database: sDocDB

Document ID: StatusNumberID

Field Name: VoiceStatus

Then Goto: Main Menu

sDocServer Defines the target database that stores the requests. This variable is defined in "Application Setup."

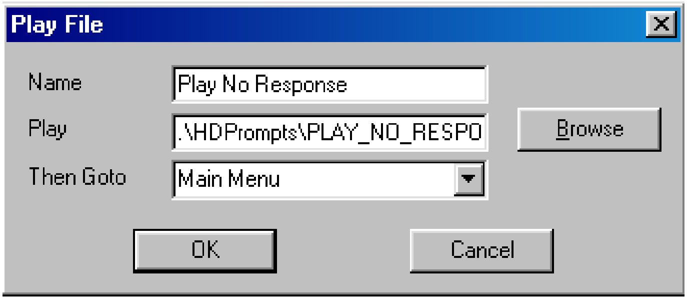
sDocDB Defines the target database that stores the requests. This variable is defined in "Application Setup."

StatusNumberID The ID for the current document, which is set in the "Look Up Status Number" branch.

Voice Status Defines the field that the script uses.

Play No Response

Plays the Play_no_response.wav file and sends the call to the main menu.



Play File

Name: Play No Response

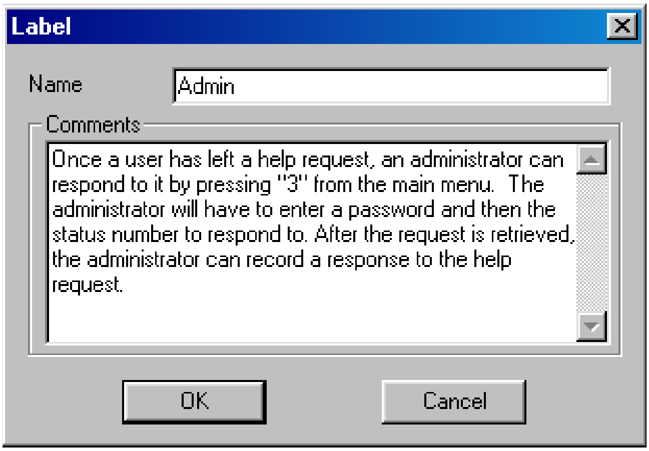
Play: .\HDPrompts\PLAY_NO_RESPO

Then Goto: Main Menu

.\Hdprompts\Play_no_response.wav The location of the prompt.

Admin

Label command that groups the administrator's tasks and options.



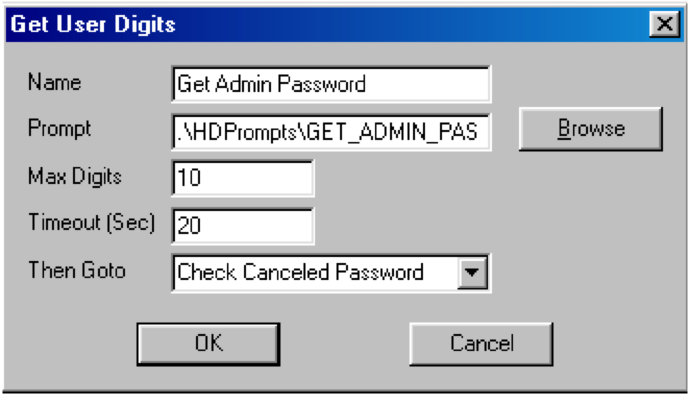
Name: Admin

Comments: Once a user has left a help request, an administrator can respond to it by pressing "3" from the main menu. The administrator will have to enter a password and then the status number to respond to. After the request is retrieved, the administrator can record a response to the help request.

Comments Stores comments about this group of commands. For reference purposes only, no action is taken.

Get Admin Password

The administrator menu items are password protected. This command validates the entered password. The password is compared to the Setup document in the HelpDesk database.



Name: Get Admin Password

Prompt: .\HDPrompts\GET_ADMIN_PAS

Max Digits: 10

Timeout (Sec): 20

Then Goto: Check Canceled Password

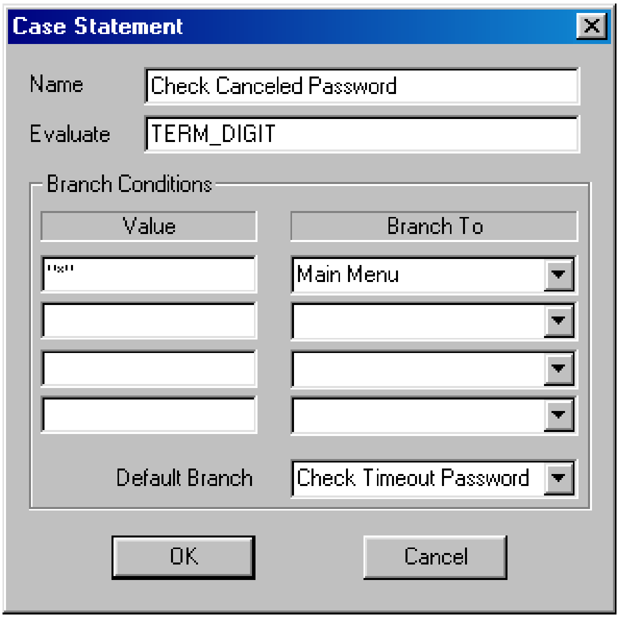
.\Hdprompts\Get_admin_password.wav The location of the prompt.

Max Digits 10 is the maximum number of digits a caller can enter.

Timeout (Sec) 20 seconds limit for a caller to enter the first digit.

Check Canceled Password

If a caller presses ✶ or cancels the password entry, this command sends the caller back to the main menu.



Name: Check Canceled Password

Evaluate: TERM_DIGIT

Branch Conditions

Value	Branch To
"*"	Main Menu

Default Branch: Check Timeout Password

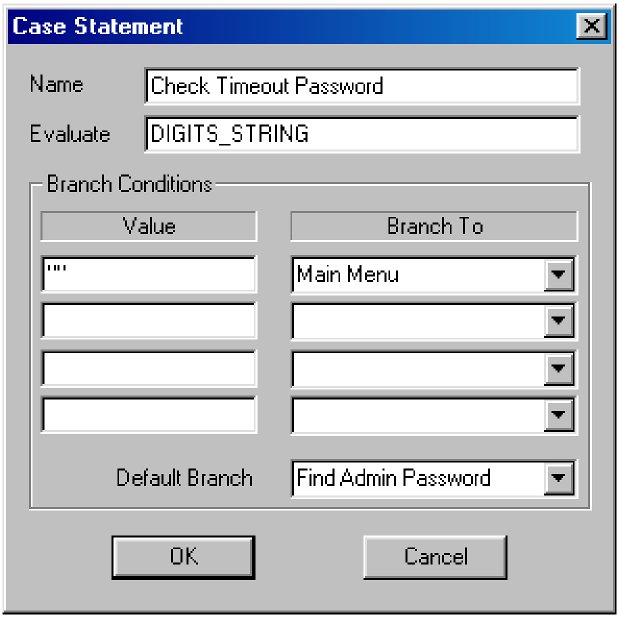
TERM_DIGIT Value evaluated for a termination character resulting from the Get Digits command.

Branch Conditions If a caller presses ✶ to exit this menu, the application branches to the main menu.

Default Branch If a caller enteres any digit other than ✶, the script branches to “Check Timeout Password.”

Check Timeout Password

Checks for a caller that hangs up or waits too long to enter a valid number. To avoid an infinite call loop, this command branches back to the main menu.



Name: Check Timeout Password

Evaluate: DIGITS_STRING

Branch Conditions

Value	Branch To
""	Main Menu

Default Branch: Find Admin Password

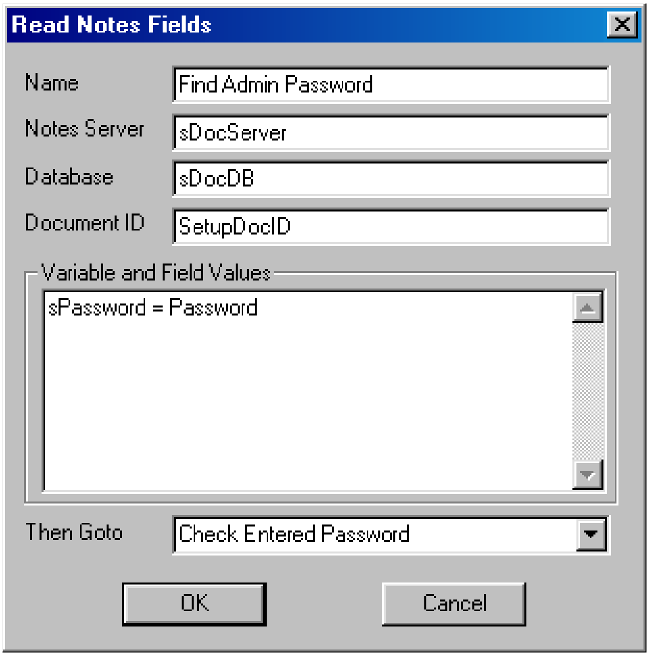
DIGITS_STRING A reserved word resulting from the Get Digitscommand.

Branch Conditions If the caller enters nothing, DIGITS_STRING is null or empty and the script returns to the main menu. The two double quotes represent an empty string.

Default Branch If the caller enters a digit, branches to “Find Admin Password.”

Find Admin Password

Retrieves the administrator password from the Setup document and stores it in the “sPassword” variable.



Read Notes Fields

Name: Find Admin Password

Notes Server: sDocServer

Database: sDocDB

Document ID: SetupDocID

Variable and Field Values

sPassword = Password

Then Goto: Check Entered Password

sDocServer Defines the server with the target database.

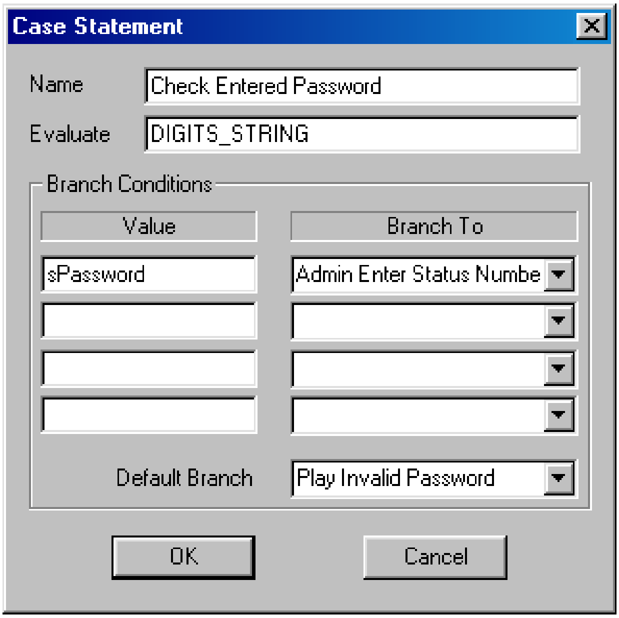
sDocDB Defines the target database.

SetupDocID The document ID created in “Find Setup Doc.”

sPassword = Password sPassword is a variable and “Password” is the field name with the password value. The “Password” field is set to “123.”

Check Entered Password

Evaluates the DIGITS_STRING reserved word from the Get Digits command and compares it to the “sPassword” variable, which is currently set as “123.”



Case Statement

Name: Check Entered Password

Evaluate: DIGITS_STRING

Branch Conditions

Value	Branch To
sPassword	Admin Enter Status Numbe

Default Branch: Play Invalid Password

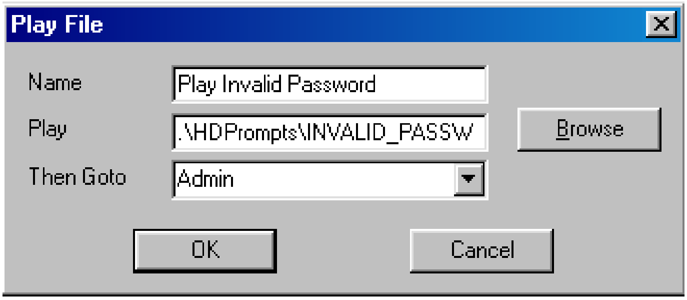
DIGITS_STRING A reserved word resulting from the Get Digits command.

Branch Conditions sPassword is compared to the DIGITS_STRING reserved word. If the password values match, the script branches to “Admin Enter Status Number Sub.”

Default Branch If the passwords do not match, the script branches to “Play Invalid Password.”

Play Invalid Password

If the DIGITS_STRING reserved word and the sPassword variable values do not match, the caller hears “I'm sorry, that is not the correct administration password.” Afterwards, the script branches to “Admin,” so the caller can try again.



Name: Play Invalid Password

Play: .\HDPrompts\INVALID_PASSW

Then Goto: Admin

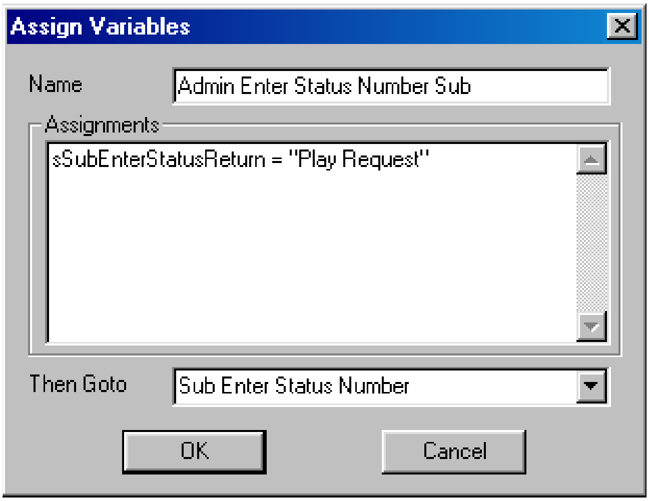
.\Hdprompts\Invalid_password.wav The location of the prompt.

Admin Enter Status Number Sub

In HelpDesk, this command is used a few times to validate the status number entered by a caller.

To prevent repeating the same commands, a subroutine is used. This command stores the "Play Request" variable in "sSubEnterStatusReturn" and then goes to the "Sub Enter Status Number" subroutine. After the subroutine is complete, the script branches to "sSubEnterStatusReturn." This variable is assigned the appropriate next branch with the "Assign variables" command.

NOTE: *You can have more than one return branch in the "Assignments" section.*



Name: Admin Enter Status Number Sub

Assignments:

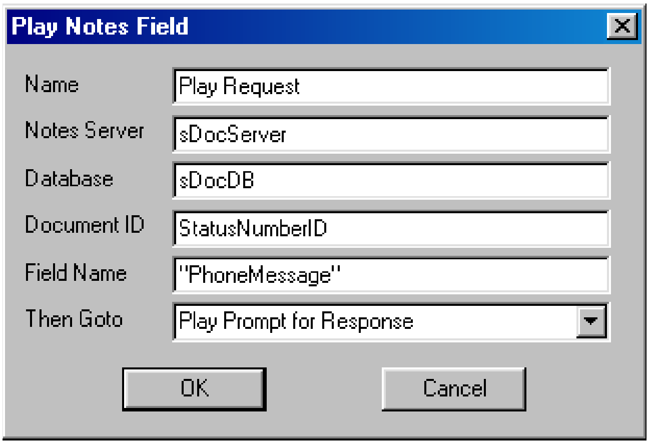
```
sSubEnterStatusReturn = "Play Request"
```

Then Goto: Sub Enter Status Number

sSubEnterStatusReturn = "Play Request" "Play Request" is assigned to the "sSubEnterStatusReturn" variable. After the "Sub Enter Status Number" subroutine runs, the script branches to "Play Request."

Play Request

After confirming the entered status number is valid, the caller hears the help request that is stored in the "PhoneMessage" field.



Play Notes Field

Name	Play Request
Notes Server	sDocServer
Database	sDocDB
Document ID	StatusNumberID
Field Name	"PhoneMessage"
Then Goto	Play Prompt for Response

sDocServer The server name that stores the target database. This and the database variables are defined in "Application Setup."

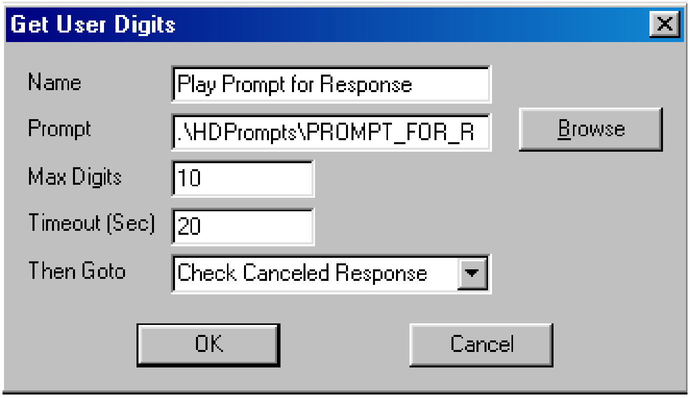
sDocDB Target database.

StatusNumberID The document ID that is created in the "Look Up Status Number" branch.

PhoneMessage Feld name with the wanted value.

Play Prompt for Response

With this branch, the caller hears “press 1 to leave a response to this request; press any other key to continue without leaving a response.”



Get User Digits

Name: Play Prompt for Response

Prompt: .\HDPrompts\PROMPT_FOR_R

Max Digits: 10

Timeout (Sec): 20

Then Goto: Check Canceled Response

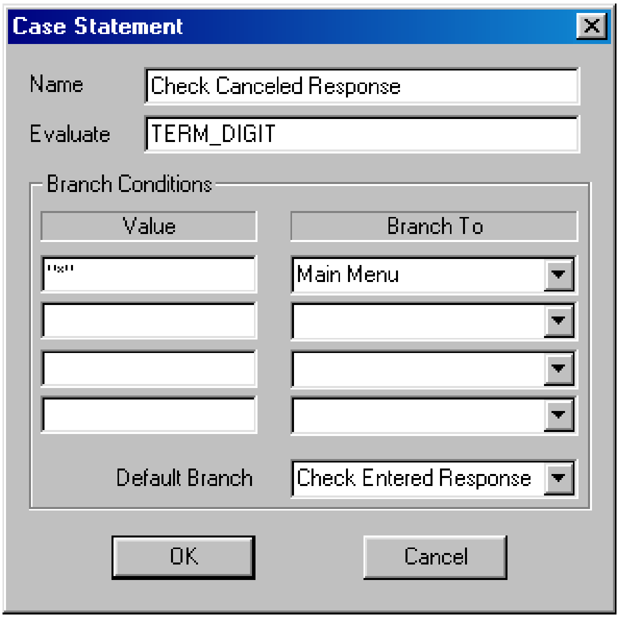
.\Hdprompts\Prompt_for_response.wav Location of the prompt.

Max Digits 10 is the maximum number of digits a caller can enter.

Timeout (Sec) 20 seconds limit for a caller to enter the first digit.

Check Canceled Response

Checks for a cancelled response. If a caller enters ✶, the script branches to the main menu. If not, the script branches to “Check Entered Response.”



Name: Check Canceled Response

Evaluate: TERM_DIGIT

Branch Conditions

Value	Branch To
"*"	Main Menu

Default Branch: Check Entered Response

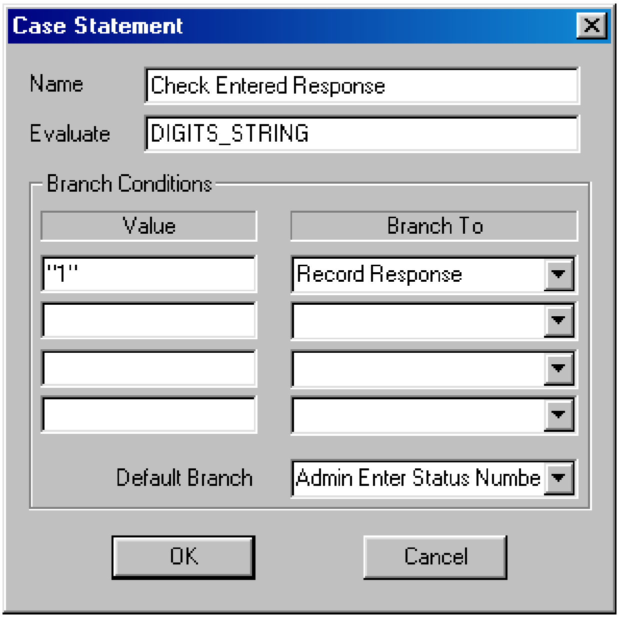
TERM_DIGIT A reserved word that acts when a caller enters ✶ or #.

Branch Conditions If the caller enters ✶, the script branches to the main menu.

Default Branch If the caller enters any other digit, the script branches to “Check Entered Response.”

Check Entered Response

Checks for whether the caller presses 1 for leaving a response. If so, the script branches to “Record Response.” If the caller presses a digit other than 1, this commands branches to “Admin Enter Status Number Sub.”



Name: Check Entered Response

Evaluate: DIGITS_STRING

Branch Conditions

Value	Branch To
"1"	Record Response

Default Branch: Admin Enter Status Numbe

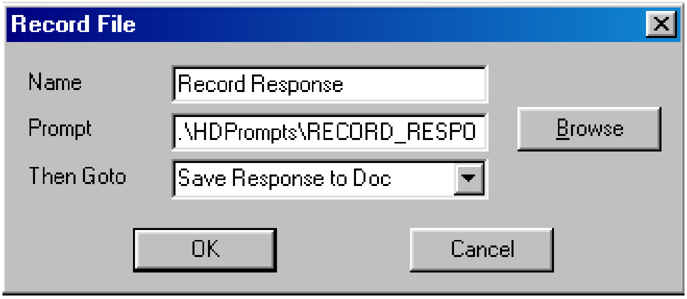
DIGITS_STRING A reserved word resulting from the Get Digits command.

Branch Conditions If a caller presses 1, branch to “Record Response.”

Default Branch If the caller presses something other than 1, branch to “Admin Enter Status Number Sub.”

Record Response

Prompts a caller to record a response. The response is saved in a temporary voice file in the VOICE_FILE variable.



Record File

Name: Record Response

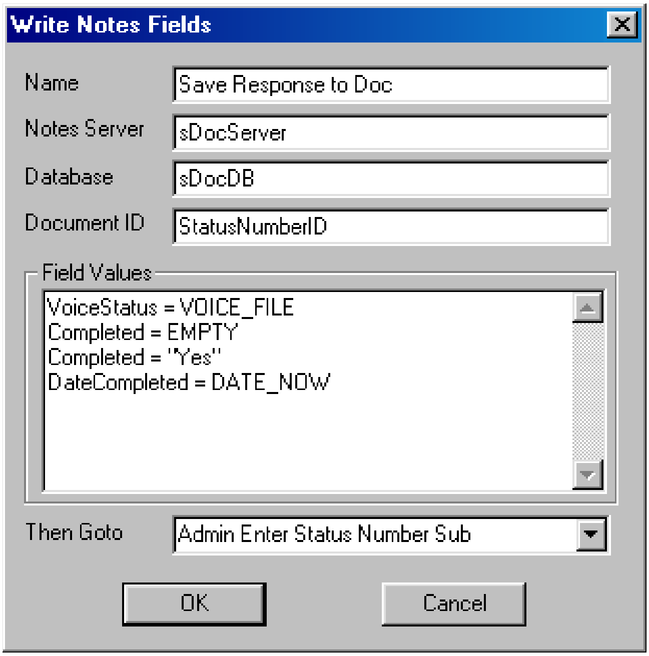
Prompt: .\HDPrompts\RECORD_RESPO

Then Goto: Save Response to Doc

.\Hdprompts\Record_response.wav Location of the prompt.

Save Response To Doc

Saves the response in the document, changes the “Completed” field to “Yes,” and sets the “DateCompleted” field to the current date and time.



Name: Save Response to Doc
Notes Server: sDocServer
Database: sDocDB
Document ID: StatusNumberID

Field Values

```
VoiceStatus = VOICE_FILE
Completed = EMPTY
Completed = "Yes"
DateCompleted = DATE_NOW
```

Then Goto: Admin Enter Status Number Sub

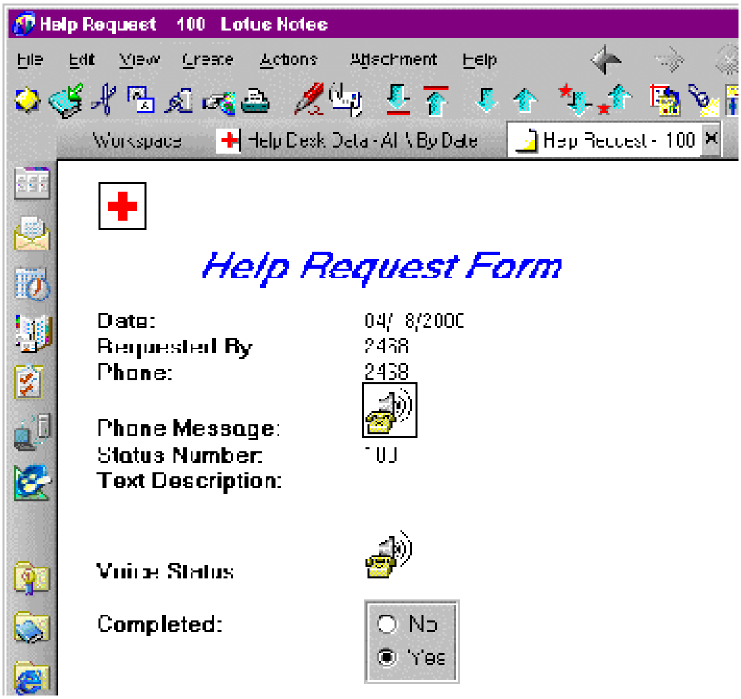
VoiceStatus = VOICE_FILE Saves the response from the VOICE_FILE reserved word to the “VoiceStatus” field in the StatusNumber ID document.

Completed = EMPTY The EMPTY reserved word is used to clear a Notes field.

Completed = “Yes” Sets the field value to “Yes.”

DateCompleted = DATE_NOW Uses the DATE_NOW reserved word to set the “DateCompleted” field to the current date and time.

Completed request



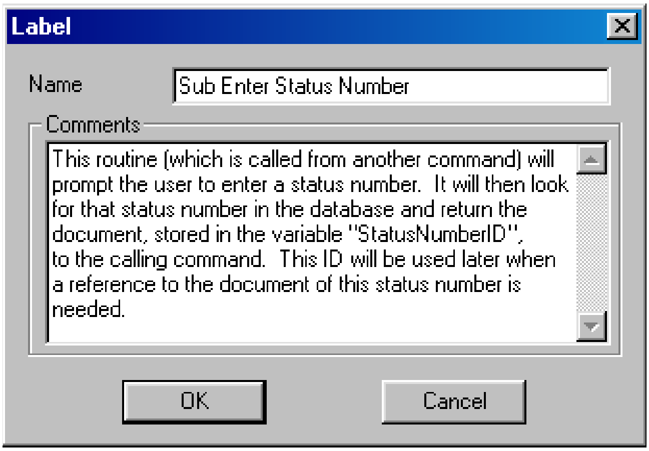
Help Request Form

Date: 04/ 8/2000
Requested By 2438
Phone: 2438
Phone Message:
Status Number: 100
Text Description:
Voice Status
Completed: ○ No ● Yes

Sub Enter Status Number

Groups the commands for a subroutine that retrieves a document based on the caller entries. If a caller enters an invalid number, this subroutine sends the call back to the beginning to try again.

This subroutine is used in the “Check Request Status” and “Admin” sections.



Label

Name: Sub Enter Status Number

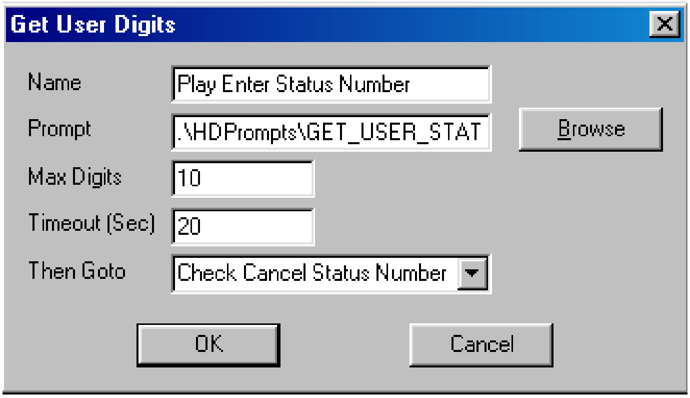
Comments: This routine (which is called from another command) will prompt the user to enter a status number. It will then look for that status number in the database and return the document, stored in the variable "StatusNumberID", to the calling command. This ID will be used later when a reference to the document of this status number is needed.

Comments Stores comments about this group. For reference purposes only, no action is taken.

NOTE: *A “Then GoTo” field is unnecessary for this command. The application branches to the next command listed.*

Play Enter Status Number

Uses the Get User Digits command to get the status number from a caller and then stores the number in the DIGITS_STRING reserved word.



Get User Digits

Name: Play Enter Status Number

Prompt: .\HDPrompts\GET_USER_STAT

Max Digits: 10

Timeout (Sec): 20

Then Goto: Check Cancel Status Number

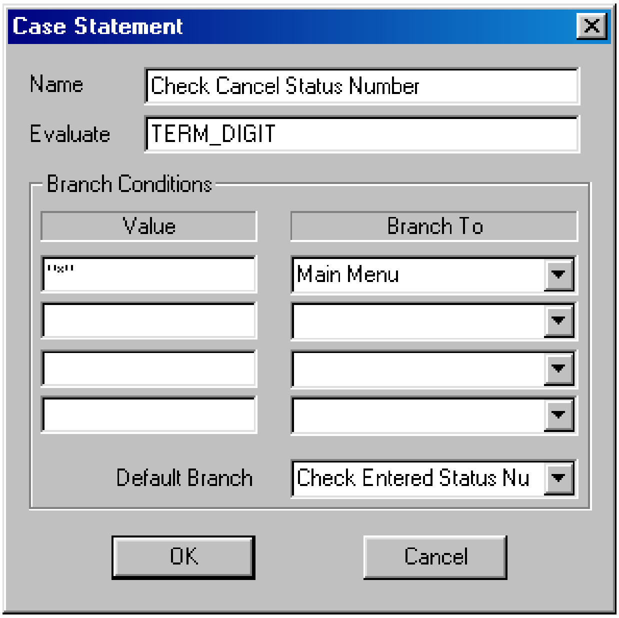
.\Hdprompts\Get_user_status.wav Location of the prompt.

Max Digits 10 is the maximum number of digits a caller can enter.

Timeout (Sec) 20 seconds limit for a caller to enter the first digit.

Check Cancel Status Number

Checks whether or not the caller presses the ✶ to cancel or hangs up during the response retrieval process.



Name: Check Cancel Status Number

Evaluate: TERM_DIGIT

Branch Conditions

Value	Branch To
"*"	Main Menu

Default Branch: Check Entered Status Nu

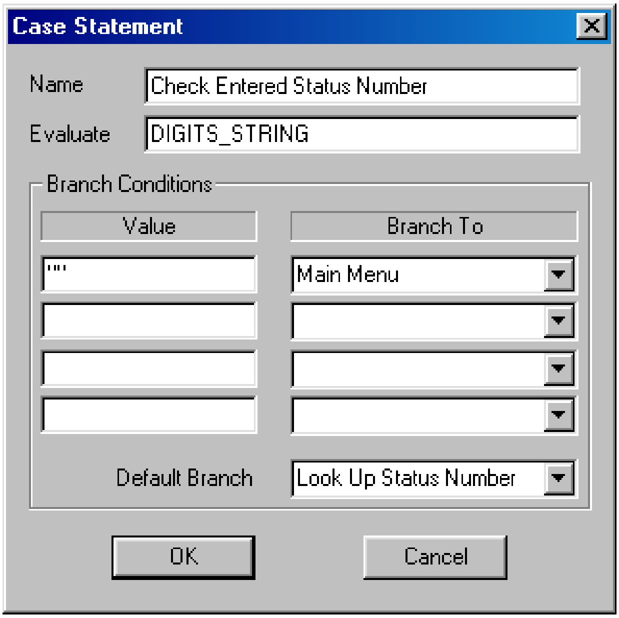
TERM_DIGIT Value evaluated for a termination character resulting from the Get Digits command.

Branch Conditions If a caller presses ✶, the application branches to "Main Menu."

Default Branch If a caller enters any digit other than ✶, the script branches to "Check Entered Status Number."

Check Entered Status Number

If the caller hangs up or waits too long to enter the first digit, this command branches to “Look Up Status Number.”



Name: Check Entered Status Number

Evaluate: DIGITS_STRING

Branch Conditions

Value	Branch To
""	Main Menu

Default Branch: Look Up Status Number

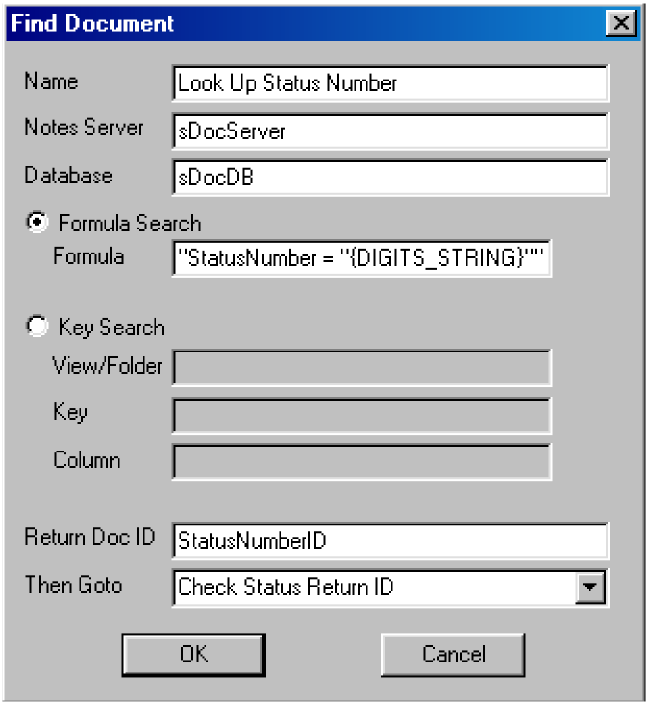
DIGITS_STRING A reserved word resulting from the Get Digits command.

Branch Conditions If a caller enters nothing or hangs up, DIGITS_STRING becomes null or empty and the script returns to “Main Menu.” The two double quotes represent an empty string.

Default Branch Otherwise, branch to “Look Up Status Number.”

Look Up Status Number

Searches for the document number entered by a caller. A formula search is used to search the “StatusNumber” field for a value equal to the DIGITS_STRING value.



Name: Look Up Status Number

Notes Server: sDocServer

Database: sDocDB

● Formula Search

Formula: "StatusNumber = "{DIGITS_STRING}""

○ Key Search

View/Folder:

Key:

Column:

Return Doc ID: StatusNumberID

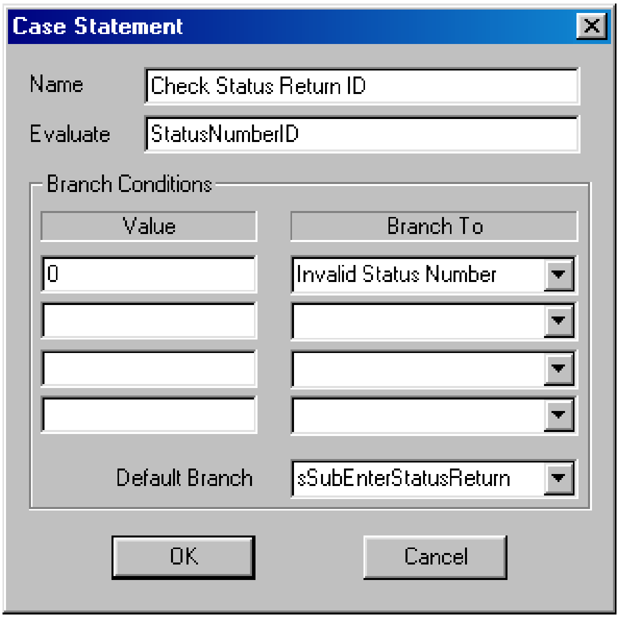
Then Goto: Check Status Return ID

“StatusNumber = “{DIGITS_STRING}”” Searches for a document whose “StatusNumber” field value is equal to the DIGITS_STRING entered by a caller.

StatusNumberID sNames the document with the matching field values.

Check Status Return ID

Checks for a valid status number ID. If a caller enters an invalid status number, the Document ID returns a “0” (zero) value.



Case Statement

Name: Check Status Return ID

Evaluate: StatusNumberID

Branch Conditions

Value	Branch To
0	Invalid Status Number

Default Branch: sSubEnterStatusReturn

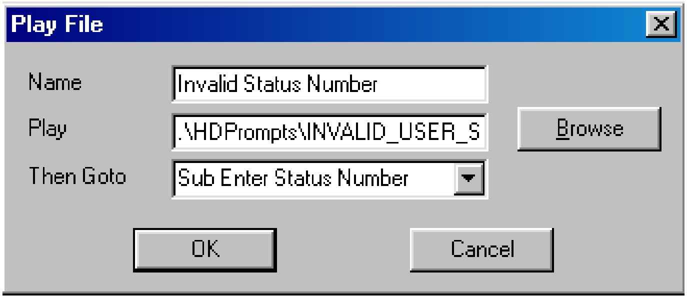
StatusNumberID The document ID from “Look Up Status Number.”

Branch Conditions If no matching documents are found, the Document ID returns a “0” value and branches to “Invalid Status Number.”

Default Branch If the value is something other than “0” (zero), branch to “SubEnterStatusReturn.”

Invalid Status Number

If the StatusNumberID returns a “0” (zero) value, the caller hears “I'm sorry, the status number you entered could not be found in the database.”



Name: Invalid Status Number

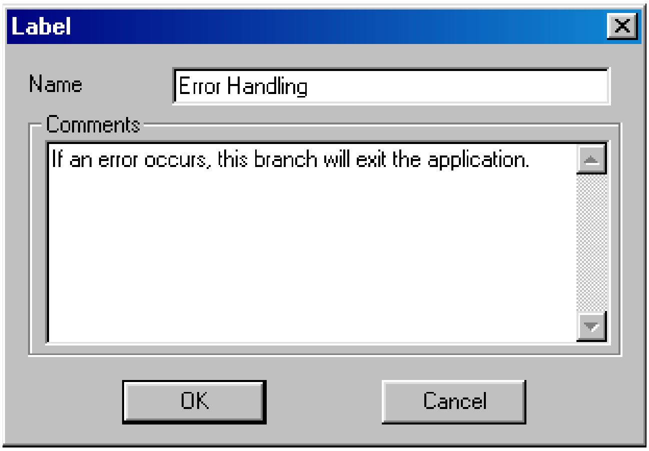
Play: .\HDPrompts\INVALID_USER_S

Then Goto: Sub Enter Status Number

.\Hdprompts\Invalid_user_status.wav Location of the prompt.

Error Handling

Label command that groups the error handling tasks and options.



Name: Error Handling

Comments: If an error occurs, this branch will exit the application.

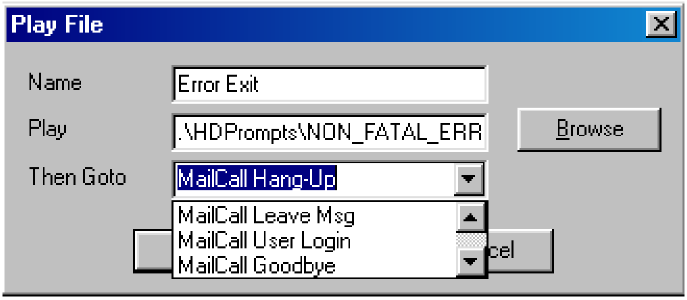
Comments Stores comments about this group of commands. For reference purposes only, no action is taken.

NOTE: *A "Then GoTo" field is unnecessary for this command. The application branches to the next command listed, which for HelpDesk is "Error Exit."*

Error Exit

If an error occurs during a call, a caller hears “I'm sorry, an error has occurred in this application; please call again later” and the application ends the call.

This command is also used in the “Check SetupIDDoc” branch. If the Setup document does not exist, the application branches here.



Play File

Name: Error Exit

Play: .\HDPrompts\NON_FATAL_ERR

Then Goto: MailCall Hang-Up

- MailCall Leave Msg
- MailCall User Login
- MailCall Goodbye

.\Hdprompts\Play_welcome.wav The location of the prompt.

Get Next Status Number

This Lotus Notes agent retrieves the current status number from the Setup document and then increments the status number in the Setup document for the next request. Refer to Notes online Help for more information on LotusScript.

NOTES:

The LotusScript commands are text and can be copied and pasted to other agents.

To use LotusScript in an agent, select "LotusScript" next to the "Run" box in the "(Agent): Action section."

To open this agent

1. From Domino Designer, open the HelpDesk database. For details, see page 72.
2. Click the View>Agents menus.
3. In the Agents section, double-click the "Get Next Status Number" agent.
4. In the bottom window of the agent, view the following sections of the LotusScript for the "Get Next Status Number" subroutine:
   - Click the "Declarations" object to view the Declarations section. The variables being assigned are: Notes Session, Database, View, Documents and Fields ("Notes Item" can be a field).
   - Click "Initialize" to view the Initialize section .

   In the following subroutine sections, the lines beginning with single quotes are comments for the following applicable line.

Get Next Status Subroutine

Declarations

```
‘Defines the “s” variable as a Notes Session.
Dim s As NotesSession
‘Defines the “Db” variable as a Notes Database.
Dim Db As NotesDatabase
'Defines the “SetupView” variable as a Notes View.
Dim SetupView As NotesView
'Defines the “ScriptDoc” variable as a Notes Document.
Dim ScriptDoc As NotesDocument
'Defines the “SetupDoc” variable as a Notes Document.
Dim SetupDoc As NotesDocument
'Defines the “NewItem” variable as a Notes Item.
Dim NewItem As NotesItem
```

Continued

Initialize variables

NOTE: *LotusScript runs this “Get Next Status Number”section when the agent is first initialized.*

```
‘Sub Initialize

'Finds the Setup Document in the hDeskDat.nsf
‘database,gets the HelpDesk Problem Report
‘Ticket number from the “NextStatus”  field,
‘assigns it to HelpDesk, and then increments
‘the “NextStatus” field value.Defines the “s”
‘variable as a new Notes Session.

Set s = New NotesSession

'Defines the “Db” variable as the current
‘database for the “s” session.

Set Db = s.CurrentDatabase

‘Defines the “ScriptDocument” variable as the
‘document within the “s” session. “Document
‘Context” stores the document in memory for
‘transferring to HelpDesk.

Set ScriptDoc = s.DocumentContext

'Defines a hidden view that lists all
‘documents by form name.

Set SetupView  = Db.GetView( “(FORMS)” )

'Locates the Setup document from the view
‘defined above.

Set SetupDoc = SetupView.GetDocumentByKey(“Setup”, True )

‘Gets the NextStatus field value from the
‘Setup document and updates the script
‘document. This value is passed back to
‘HelpDesk.

Set NewItem = SetupDoc.GetFirstItem(“NextStatus”)

Set NewItem = ScriptDoc.ReplaceItemValue(“NextStatus”, NewItem.Values(0))

'Changes existing “NextStatus” text to a
‘number for incrementing. Then converts the
‘incremented “NextStatus” number back to a
‘string and saves the document. “Cdbl”
‘converts a value to a number and “Cstr”
‘converts a value to text.

Set NewItem = SetupDoc.ReplaceItemValue(“NextStatus”,
Cstr(Cdbl(NewItem.Values(0)) + 1))
```

```
Call SetupDoc.Save( True, False )

'Ends of the “Get Next Status Number” agent initialization subroutine

End Sub
```

Configure setup agent

This Lotus Notes agent prevents the creation of more than one Setup document. For the agent to be effective, the Setup form is configured to remove the Setup document from the "Create" menu and a "hidden" view is included. After the Setup document is removed, you can create a new one by selecting the "Configure Setup" agent in the "Actions" menu.

To open this agent

1. From Domino Designer, open the HelpDesk database. For details, see page 72.
2. Click "Forms" in the Recend Databases list.
3. In the right pane, double-click "Setup."
4. In the Setup form window, right-click and select "Form Properties."
5. In the first tab of the Form Properties dialog box, confirm that the "Include in Menu" check box is cleared. This setting prevents a user from creating a Setup document with the "Create" menu.
6. Click the View>Agents menus.
7. In the Agents section, double-click the "Configure Setup" agent.
8. In the top right pane, confirm that the "Manually From Actions Menu" is selected as the "When should this agent run?" setting.
9. Confirm that the "Which document(s) should it act on?" field is set to "Run once (@Commands may be used)." This setting has the agent run when it is selected from the "Actions" menu.
10. To create a hidden view for running the Configure Setup agent, click "Views" in the Recent Databases List.
11. Click "New View" and name the view "(FORMS)" and then in the first column, right-click and select the "Properties" menu. The parentheses make it hidden.
12. Click "Ascending" and then click "Close."
13. In the bottom, right pane of the column value, click the Field option button and select "Form" from the list below it.
14. Save and close the view.

15 In the bottom window of the agent, view the following sections of the LotusScript for the “Configure Setup” subroutine:

- Click the “Declarations” object to view the Declarations section. The variables being assigned are: Notes Session, Database, View, Documents and Fields (“Notes Item” can be a field).
- Click “Initialize” to view the Initialize section .

In the following subroutine sections, the lines beginning with single quotes are comments for the following applicable line.

Configure Setup Subroutine

Declarations

```
'Defines the “s” variable as a Notes Session.

Dim s As NotesSession

'Defines the “db” variable as a Notes
‘Database.

Dim db As NotesDatabase

'Defines the “view” variable as a Notes View.

Dim view As NotesView

'Defines the “setupDOC” variable as a Notes
‘Database.

Dim setupDoc As NotesDocument

'Refers to the current location where the
‘agent is running.

Dim uiWork As NotesUIWorkspace

'Refers to the document currently open in
‘Notes.

Dim uiDoc As NotesUIDocument
```

Continued

Initialize variables

NOTE: *LotusScript runs this section when the agent is first initialized.*

```
Sub Initialize

'Defines the “s” variable as a new Notes
‘Session

Set s = New NotesSession

‘Defines the “uiWork” variable as a new Notes
‘UI workspace.

Set uiWork = New NotesUIWorkspace

'Defines the “Db” variable as the Current
‘Database of the “s” Notes session

Set db = s.CurrentDatabase

‘Finds the Setup document in “Forms”

Set view = db.GetView( “(FORMS)” )

Set SetupDoc = view.GetDocumentByKey( “Setup”, True )

'If Setup document exists, edit it. If not,
‘create a new document.

   If (SetupDoc Is Nothing) Then

      Set uiDoc = uiWork.ComposeDocument( ““, ““, “Setup” )

'If the Setup document does exist, open it
‘in “Edit” mode.

   Else

      Set uiDoc = uiWork.EditDocument( True, SetupDoc, False )

   End If

'Ends the initialization subroutine

End Sub
```

CHAPTER 4: FaxBack application

Setting up FaxBack

The FaxBack application creates and sends outgoing fax messages to a fax gateway. If a configured fax gateway is unavailable or you want to receive and view the faxes locally, perform the first procedure below to set up Notes for sending fax messages to a local mailbox.

Perform the second following procedure to set up FaxBack on Phone line 1. The demonstration software can run one application at a time, because Phone line 1 is the only available line. Phone lines 2 and higher are unavailable.

NOTE: *For details on setting up Notes or a fax gateway, see your Notes mail and Notes network configuration documentation.*

To set up Notes for sending faxes to a local mailbox

1. In a local Notes mailbox, click the File>Mobile>Edit Current Location menus.
2. Click the Mail tab and confirm that the “Mail file location” is set to local, and the “Mail file” is set to a valid local mail file.

To set up Phone Server to start FaxBack

1. In the FaxBackDemo directory, locate and copy the FaxBackLib.nsf file to your Notes Data directory.
2. Start Phone Server.
3. Confirm the activity for Phone line 1 is blank or inactive. If active, click the Activity field for Phone line 1 and then click the “Stop all” tool bar button.
4. Double-click Phone line 1 and the select the FaxBack application from the “Application” list.
5. In the Line Setup dialog box, confirm that the “Autostart” check box is selected and then click “Start.”
6. Click “Close” and confirm that FaxBack is running on line 1.

Requesting and viewing faxes

After setting up Notes and Phone Server for the FaxBack application, perform the following to send a fax.

FaxBack supports the standard Lotus fax servers and other third party fax servers that have a Notes gateway interface.

The requested fax documents are sent to the fax gateway at the “Fax” domain. The mailing address is FaxBackcaller@<fax number entered>@ FAX.

Depending on how your Notes mail options and Notes network are configured, one of the following occurs when you request a fax:

- If a Notes compatible fax gateway is installed and set up with the domain name FAX and Phone Server is connected to a mail server, requested faxes are sent to the specified fax number.
- If Phone Server is connected to a mail server, but no fax gateway is set up, requested faxes are sent to your Notes mailbox with a Delivery Failure heading.
- If Phone Server is not connected to a mail server, requested faxes are stored in your Mail.box database or local mail file.

To send a fax

1. From Phone Server, simulate a caller by clicking the FaxBack application on line 1 and then click the “Demo Phone” toolbar button.
2. When asked, use the Demo Phone keypad to enter your fax number followed by #.
3. When asked, enter the document number that you want to fax. For example, press 000 to receive a catalog of all available documents.
4. The system sends the requested fax to the fax gateway that is set up in Notes. When asked for another document number, you can hang up or enter another number.

Continued

To view a fax in your mailbox or mail file

1. Open your mailbox.
2. If the requested fax was sent to your Notes mailbox, go to step 3.

   If the fax was sent to your Mail.box database, open the database, locate and cut the message. Then paste the message in your local mail file.

   NOTE: *To open the Mail.box database, you may need to add it to your desktop.*
3. Locate the message in your mailbox or mail file by looking for the following:
   - A message from the Notes mail router or from yourself.
   - A message with a subject that begins with “Delivery failure” or “Requested documents.”
4. Open the document and ignore any delivery failure messages.

Adding or modifying faxes

FaxBack includes a group of sample fax documents. You can view a document list by opening the database or by requesting the “000” fax document, which is the FaxBack catalog.

The FaxBack documents are in the “Faxbacklib.nsf” file, which is located in the directory that you specified during installation. The default directory is C:\Phonesoft\FaxBack.

Perform the following to open and modify the FaxBack database.

To open and add or modify fax documents

1. In Notes, click the File>Database>Open menus.
2. In the Open Database dialog box, click “Browse” and locate the FaxBack Sample database. The default directory is C:\PhoneSoft\FaxBackSample\FaxBackLib.nsf.
3. Click “Open.” If “Open” is not available, click “Add,” “Done,” and then “Open.”
4. Click the “By document number” view.

   You can view and modify documents as needed or add new documents using the Create>Document menus.

   For new documents, enter a number in the “Document Number” field, so callers can select the document. You may also want to add any new documents to the FaxBack catalog, document 000.

Glossary

. Standard symbol used in script commands to locate the specified file in the directory of the currently running application. Phone Server lists the currently running application in the “Application” list.

For example, if you run the HelpDesk application from the default C:\PhoneSoft\HelpDeskSample directory and type `.\Hdprompts\Main_menu.wav` in a script command, then the application looks for that file in the C:\PhoneSoft\HelpDeskSample directory.

.. Symbol used in script commands to locate the specified file in the directory one level up from the currently running application. In the example above, if you included the additional. (period) then the script looks in the C:\Phonesoft directory.

ANI (automatic number identification) The detection of the digits in a calling phone number.

Character string A group or sequence of characters.

DID (direct inward dialing) A phone system function in which calls reach specific extensions without going through an attendant.

DNIS (dialed number identification service) On an 800 or 900 line, the detection of the dialed number.

DTMF (dual tone modulation frequency) Phone system signaling in which standard pairs of specific voice band frequencies are used; touchtone dialing.

NAB Abbreviation for name and address book, which is a type of Lotus Notes database.

OLE (object linking and embedding) A way of connecting a file with objects from other applications.

SDK Abbreviation for the PhoneSoft software development kit.

Termination character A character, such as # or ✶ that is used to end the call or end the menu action. Used in the Get Digits command or a Record file.

TTS Abbreviation for text to speech, which is a technology that converts textual information into speech output.

Variable A named storage location containing data that can be modified while a program is running.

VOX File extension of a sound file.

WAV File extension of a sound file.

Index